

The Jini Technology

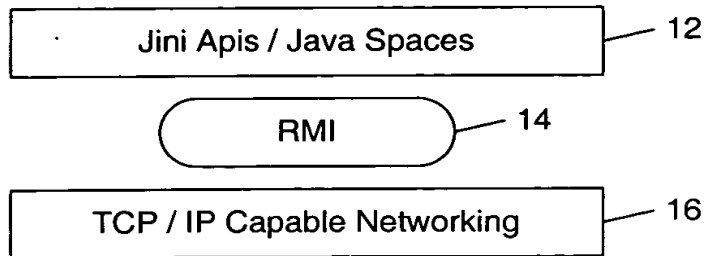


Fig. 1
(Prior Art)

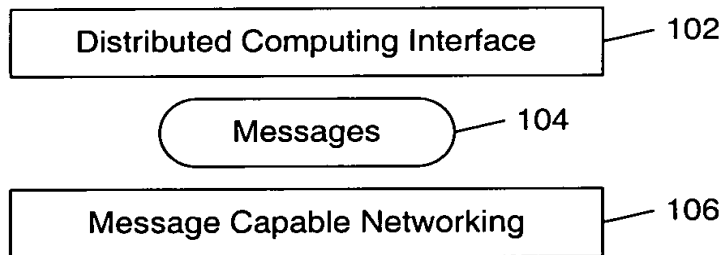


Fig. 2

00693297-101900

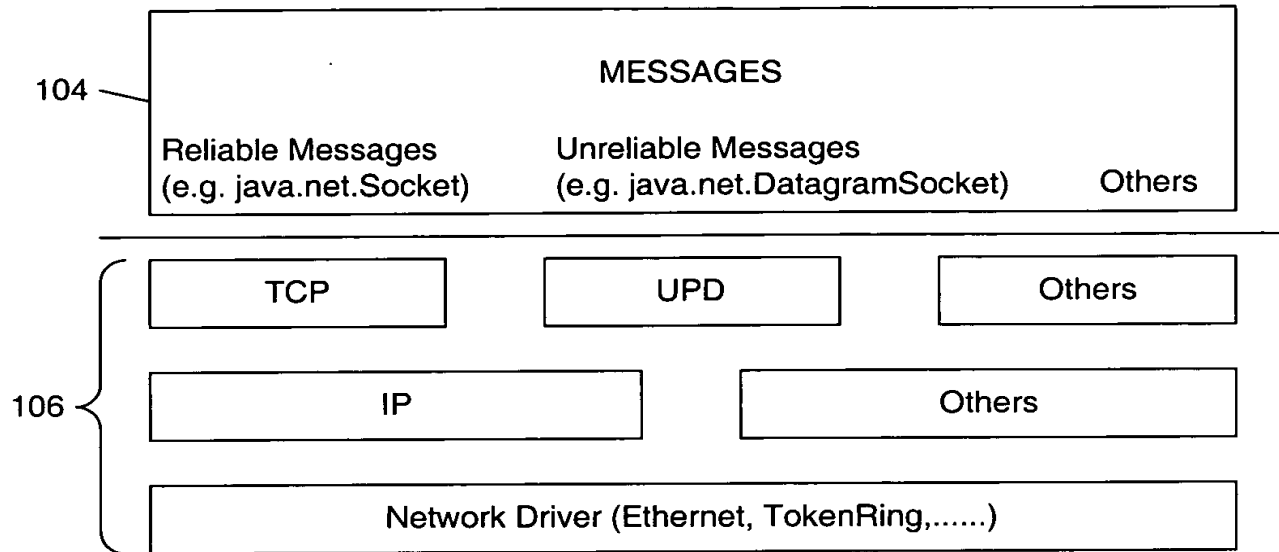


Fig. 3

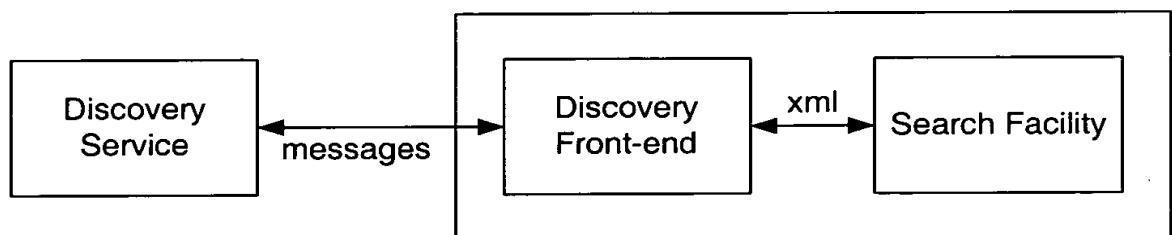


Fig. 4

00603297-101000

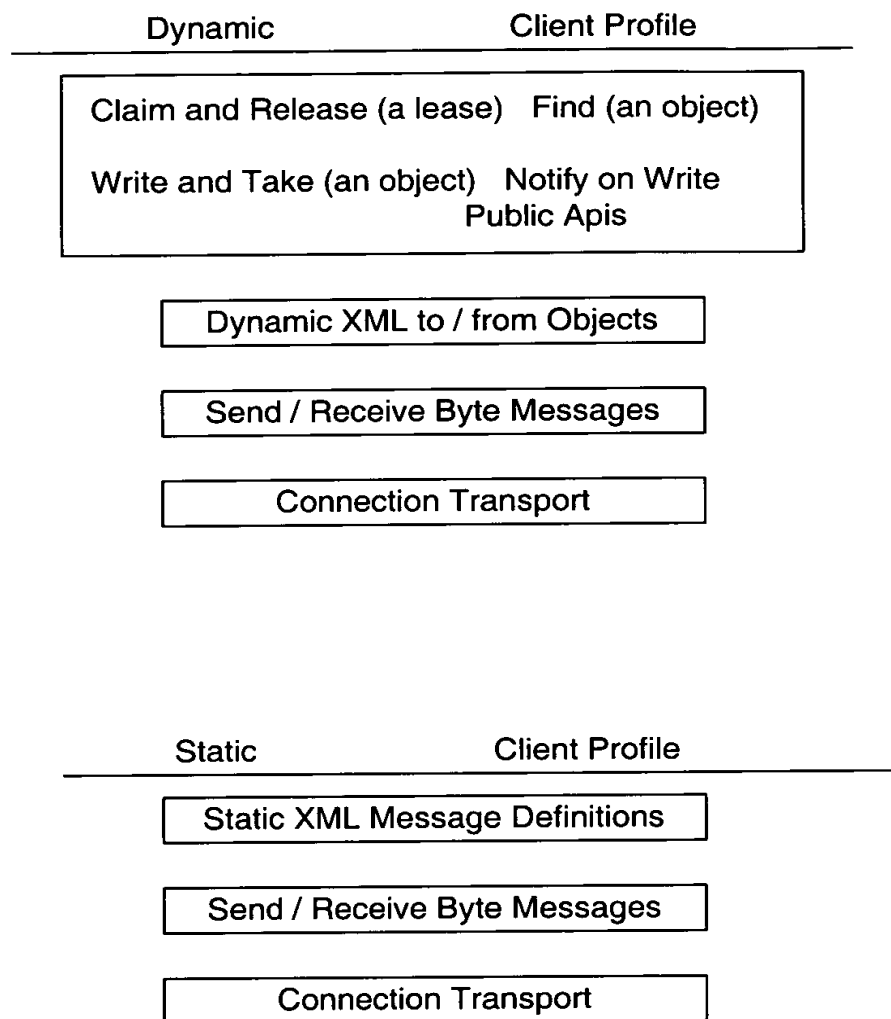


Fig. 5

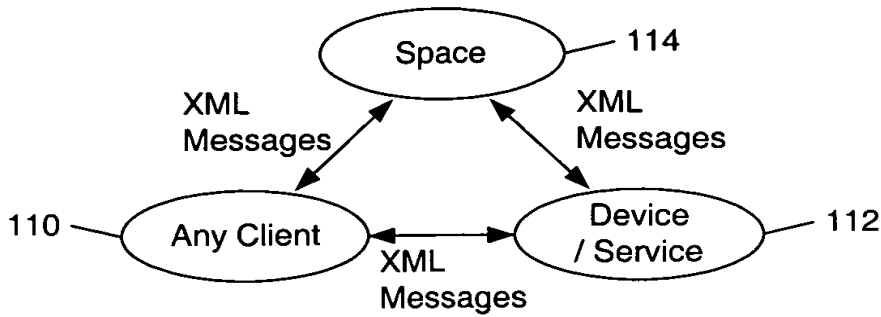


Fig. 6

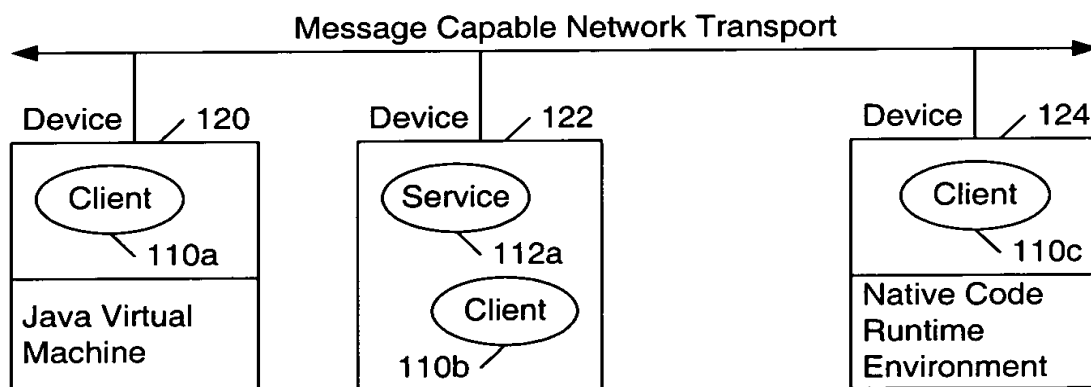
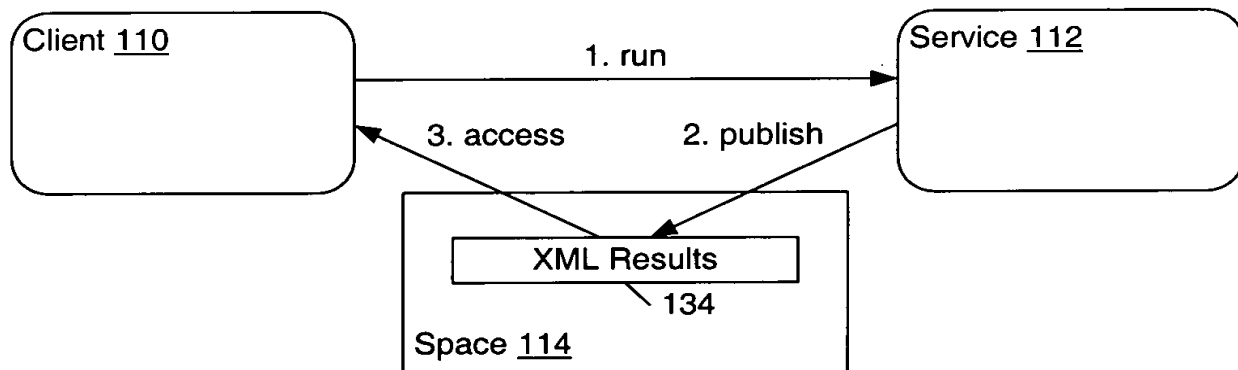
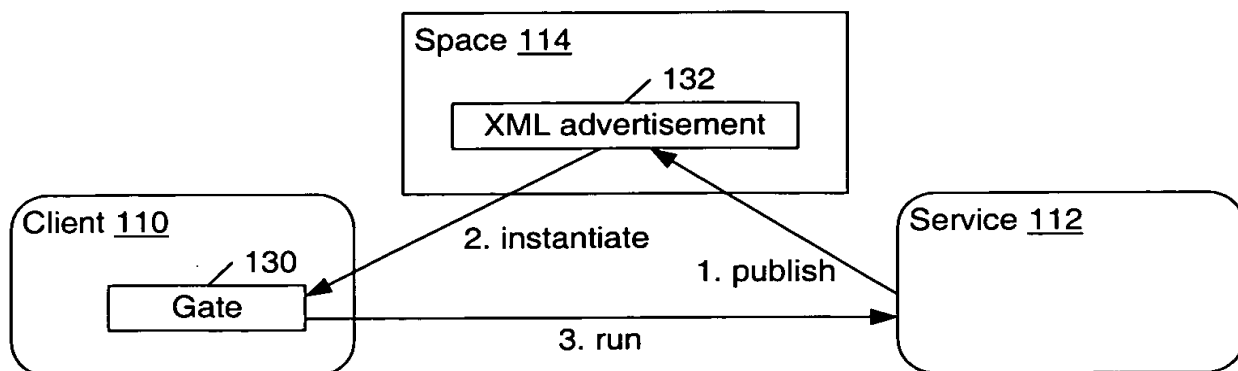
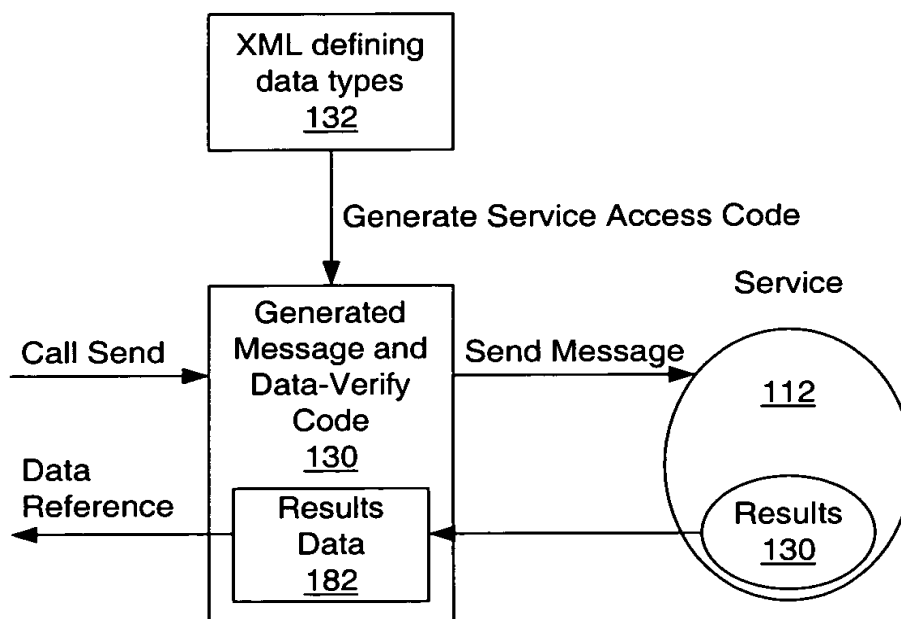
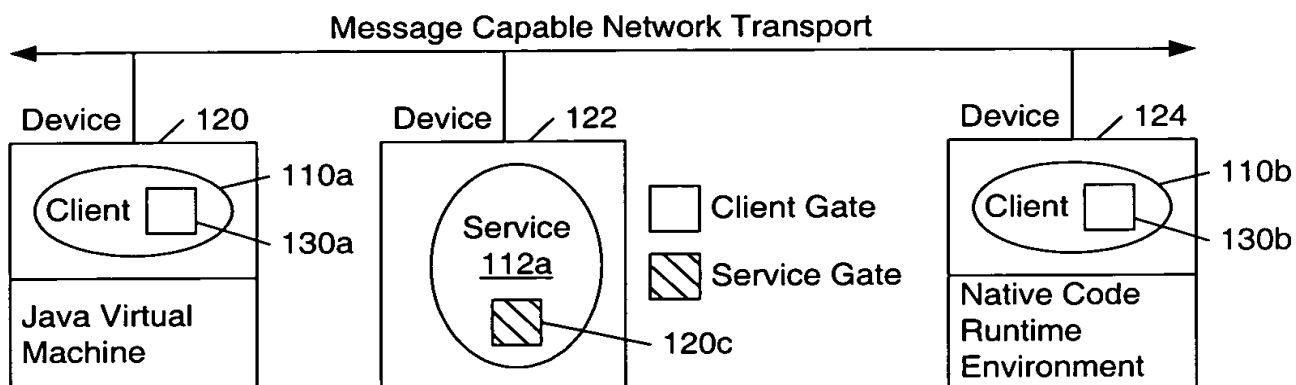
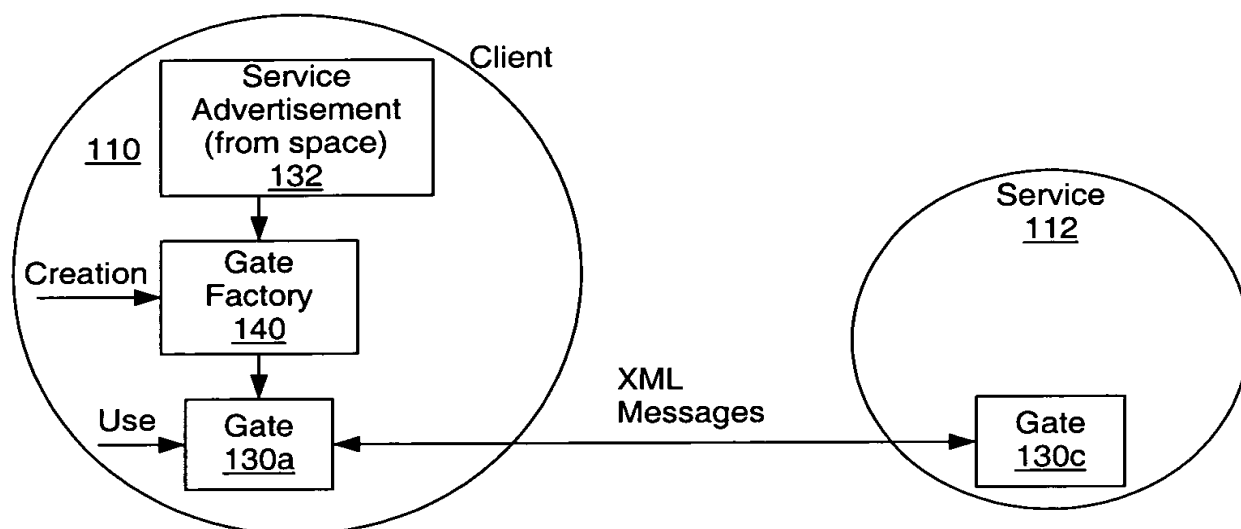
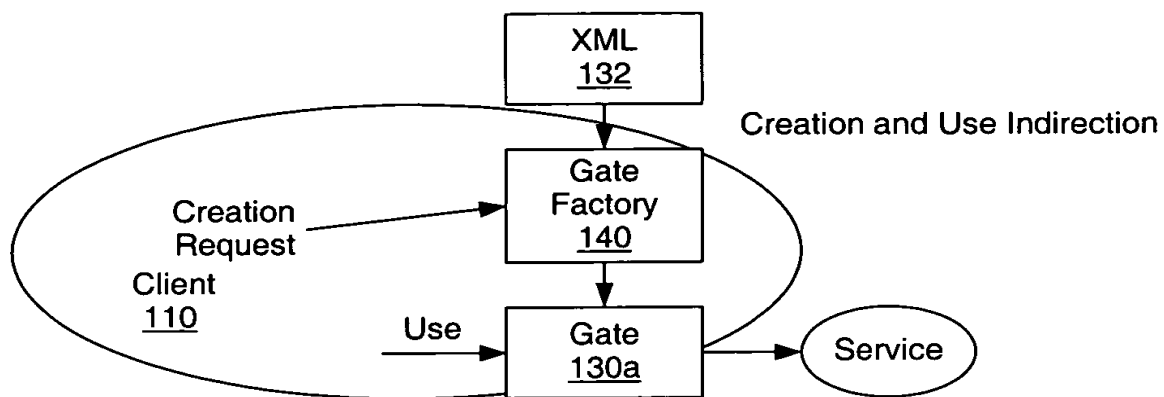


Fig. 7







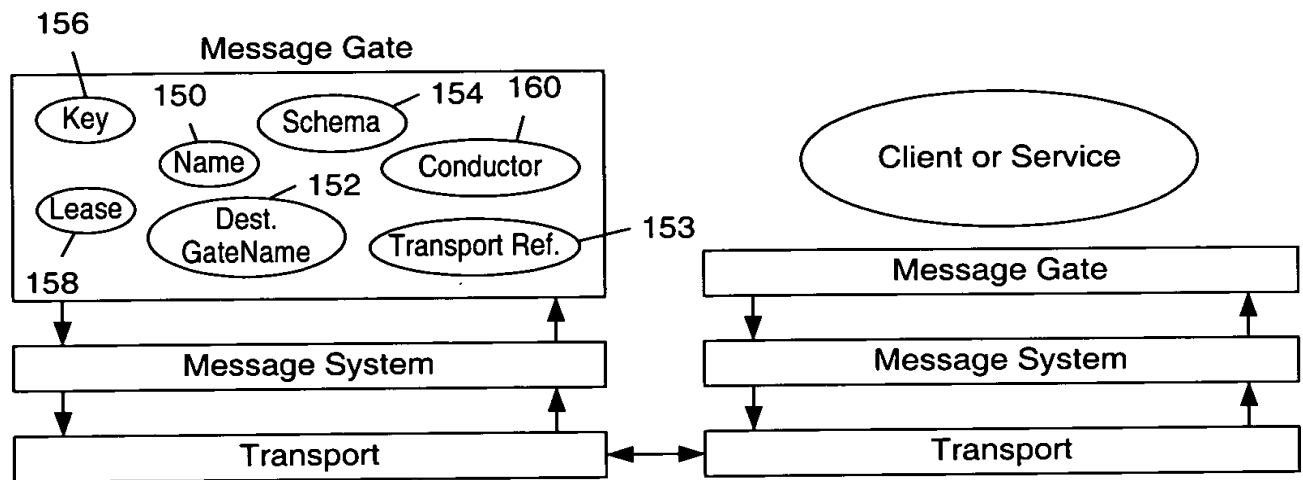


Fig. 12

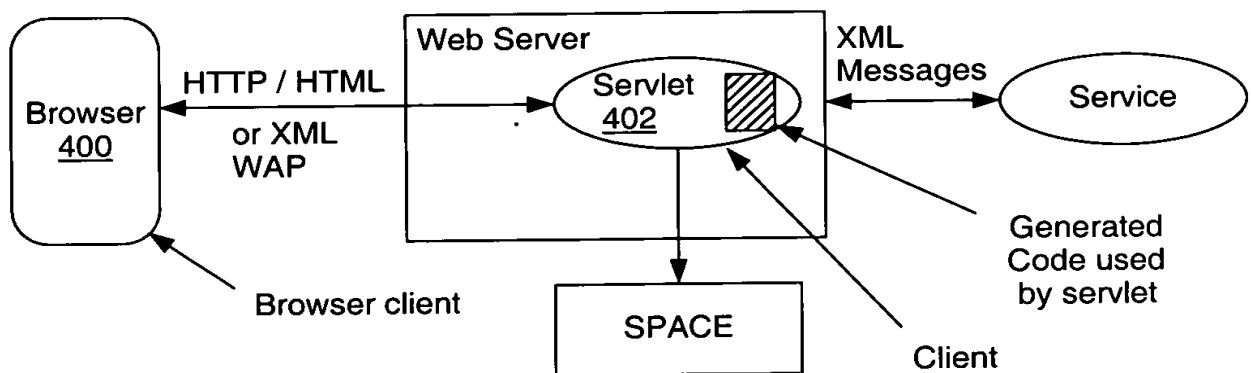


Fig. 13


```
graph TD; 170[XML defining method interface 170] -- "Generate Service Access Code" --> 172[Generated Simple Method Code 172]; 172 -- "Call Method" --> 176((Client Application 176)); 176 -- "Generated Proxy Reference" --> 172; 172 -- "Send Message" --> 180((Service 180)); 180 -- "Send Results URL" --> 172; subgraph 172 [Generated Simple Method Code 172]; 178[Results Proxy (Generated) 178]; end
```

The diagram illustrates the system architecture. At the top is a box labeled "XML defining method interface 170". An arrow labeled "Generate Service Access Code" points down to a central box labeled "Generated Simple Method Code 172". Inside this central box is a smaller box labeled "Results Proxy (Generated) 178". To the left of the central box is an oval labeled "Client Application 176". An arrow labeled "Call Method" points from the Client Application to the central box. An arrow labeled "Generated Proxy Reference" points from the central box back to the Client Application. To the right of the central box is a large oval labeled "Service". An arrow labeled "Send Message" points from the central box to the Service. Inside the Service oval is a smaller oval labeled "Results 180". An arrow labeled "Send Results URL" points from the Results 180 oval back to the central box.

```

graph LR
    Clients110[Clients 110] -- "Find advertisements" --> Space114[SPACE Filled with Service, Client, or Content Advertisements 114]
    ServiceProviders112[Service Providers 112] -- "Add advertisements" --> Space114
    Clients110 <--> |"Messages"| ServiceProviders112
  
```

Fig. 15

006T01 101900

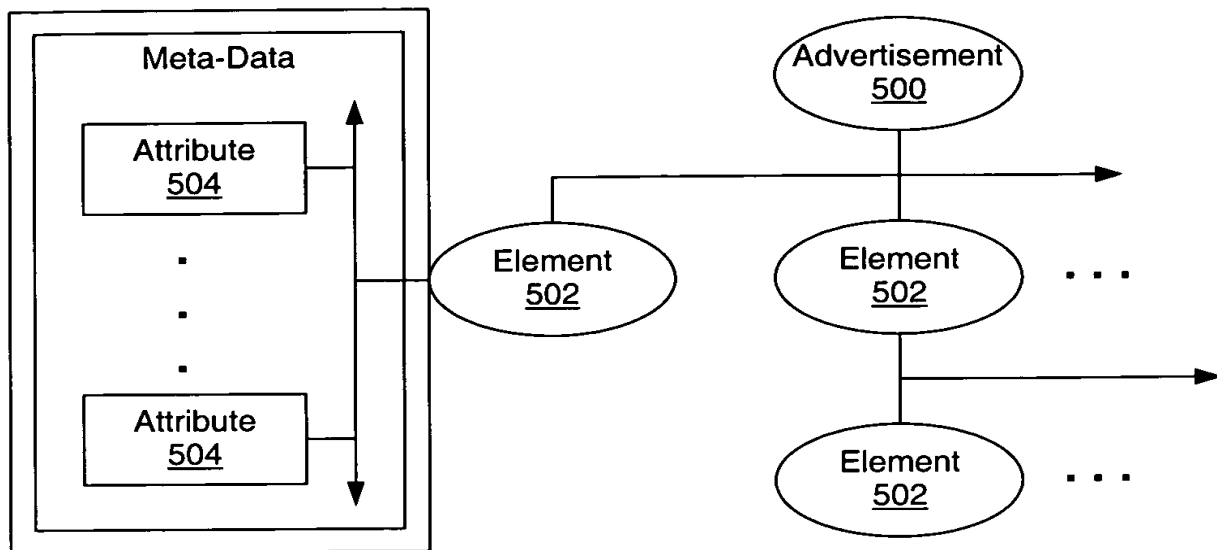


Fig. 16

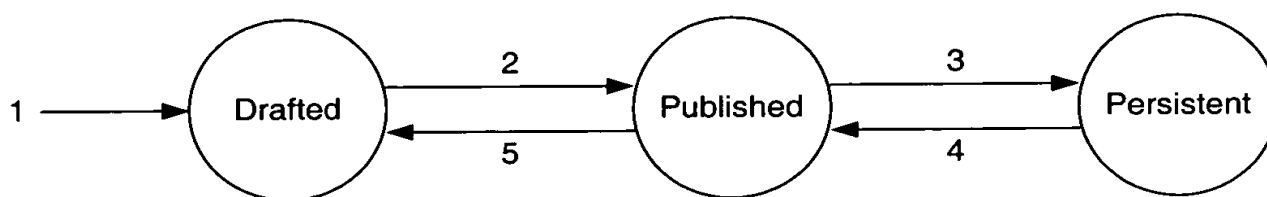


Fig. 17

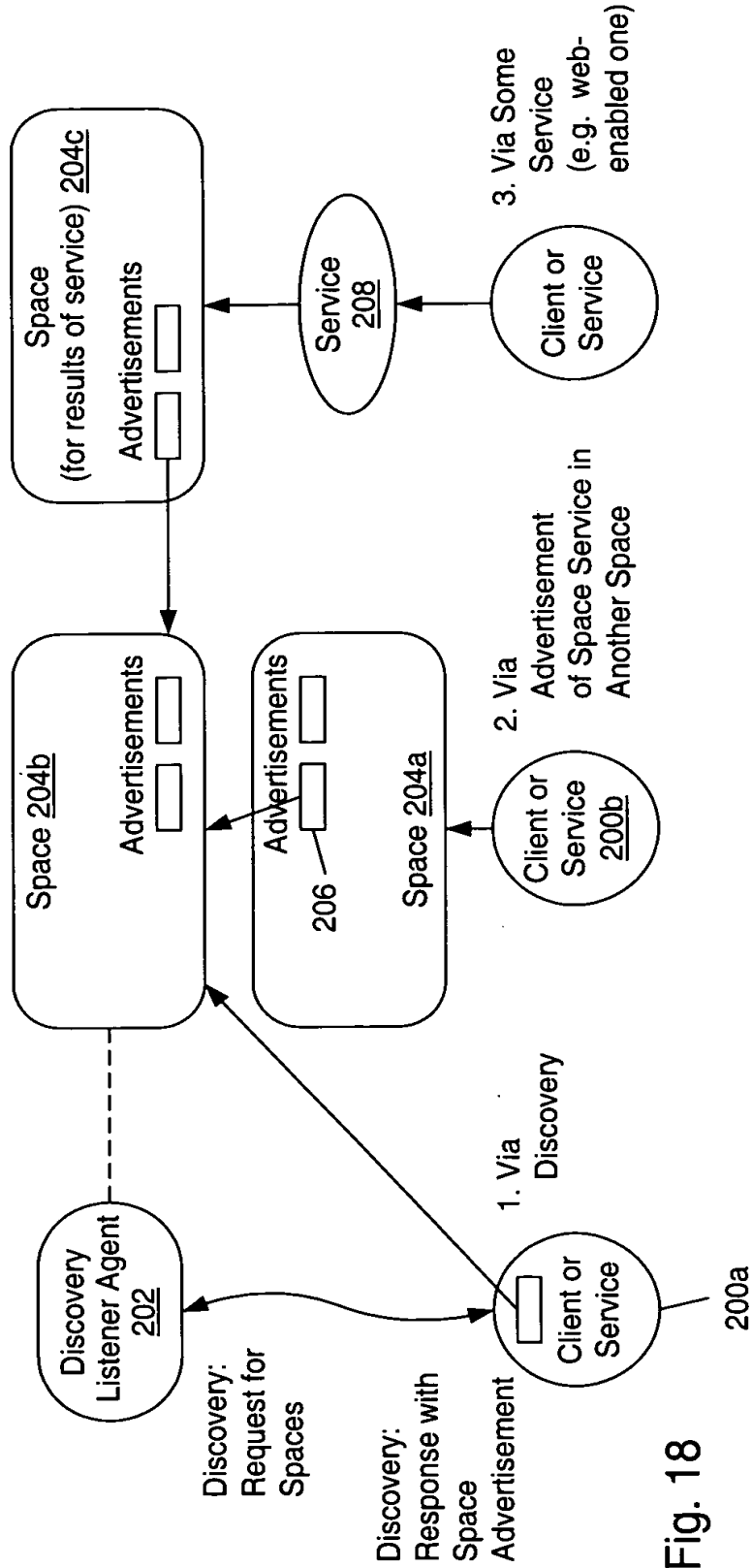


Fig. 18

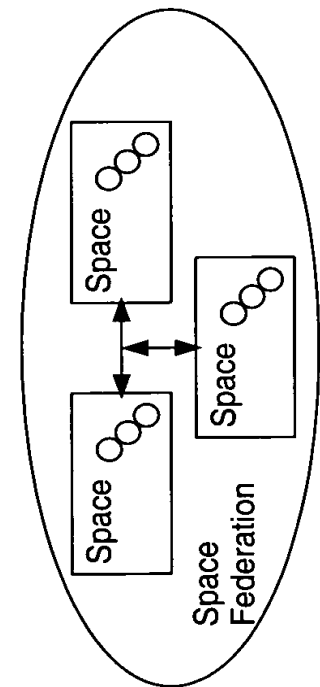


Fig. 19

```

graph TD
    300[Client runs authentication service for space  
300] --> 302[Client constructs gate for accessing space  
302]
    302 --> 304[Client sends message to space to  
run space service  
304]
    304 --> 306[Space runs authentication service to  
identify client  
306]
    306 --> 308[Space determines client capabilities and  
binds them to client identity (token)  
308]
    308 --> 310[Messaging proceeds between client and  
space - client may access space facilities  
310]

```

Fig. 20

DOT: 20200900

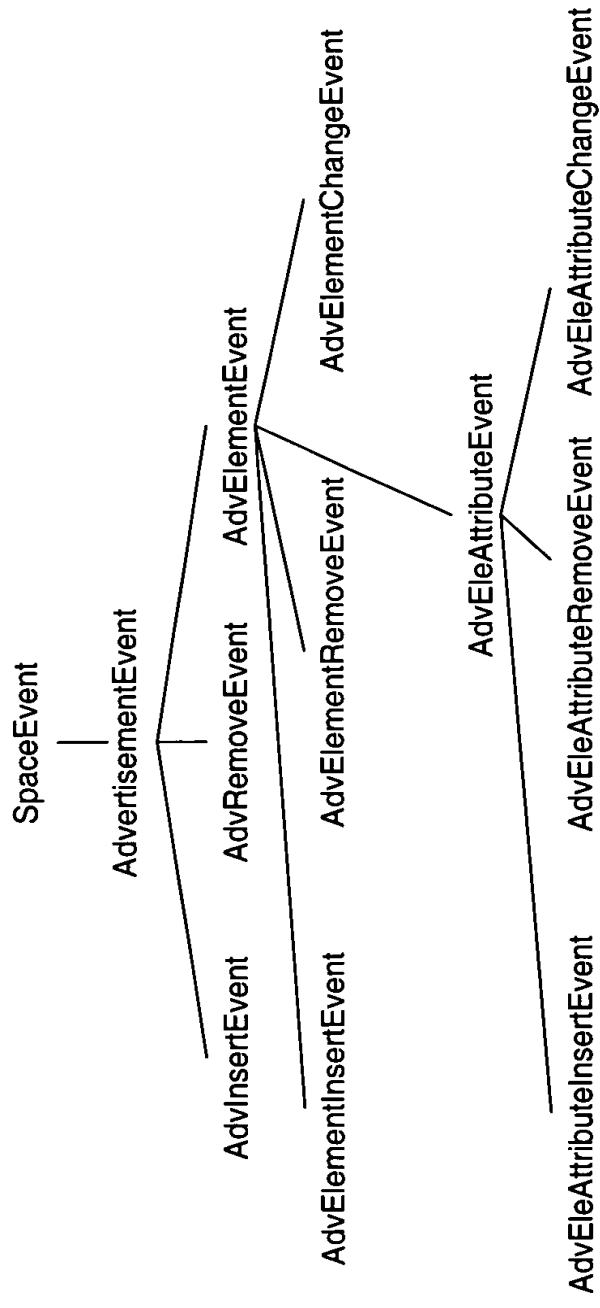


Fig. 21

006707-101000

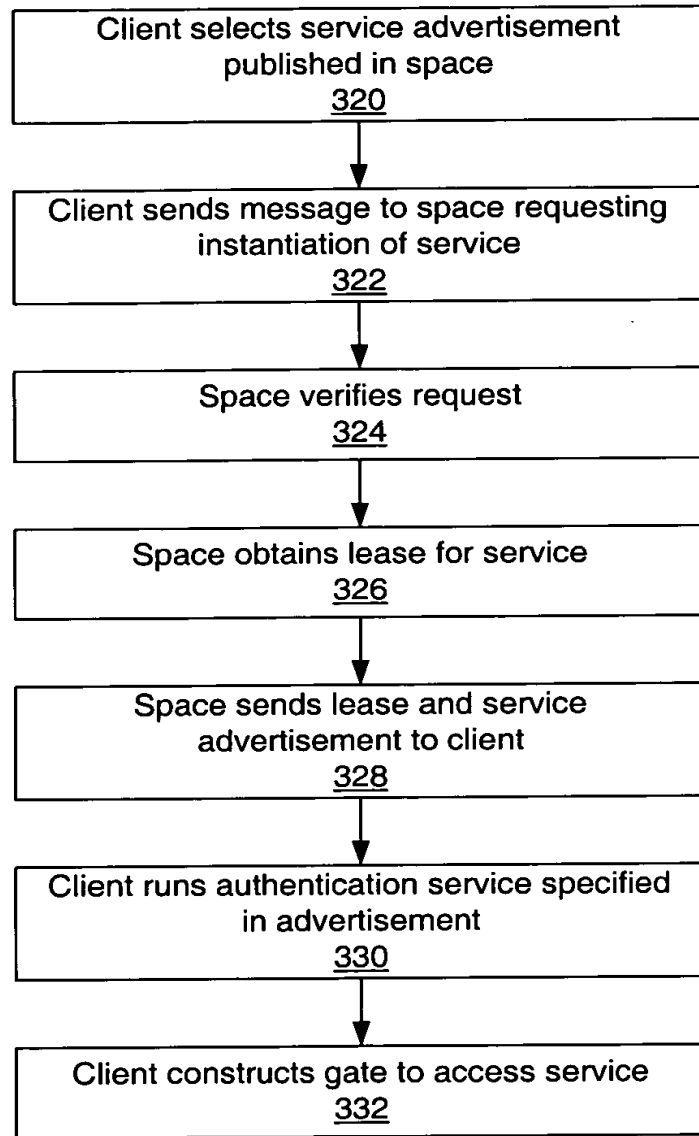


Fig. 22

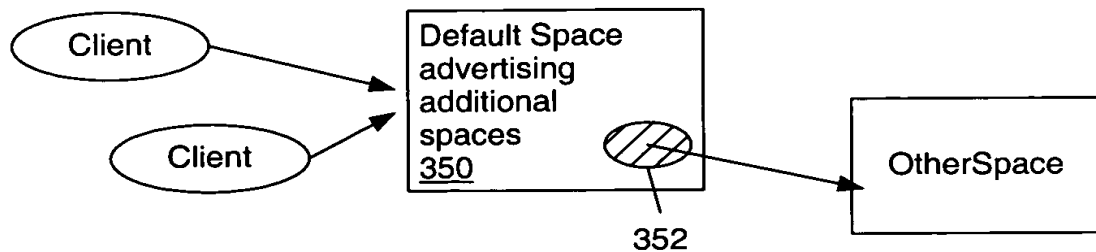


Fig. 23

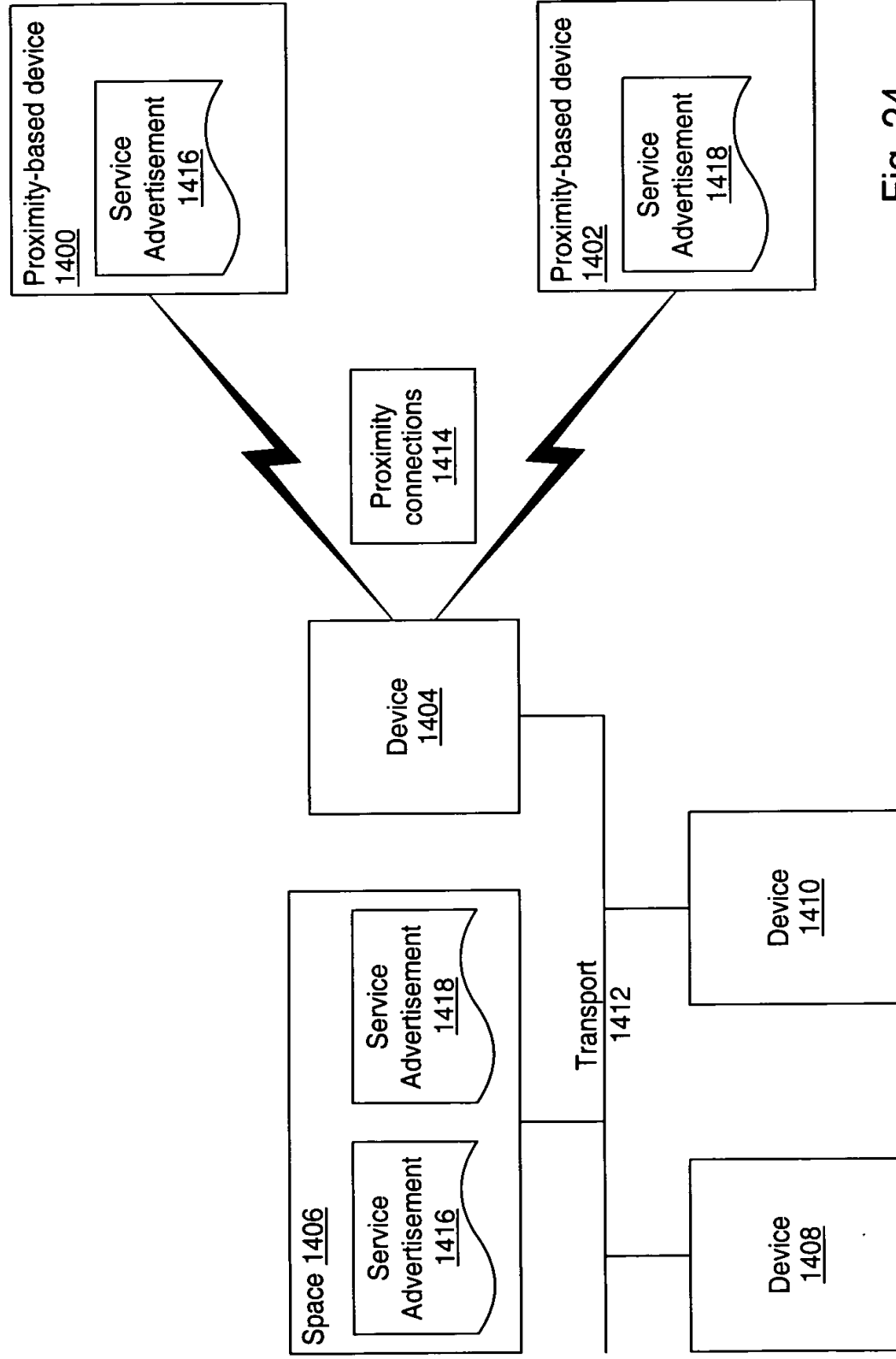


Fig. 24

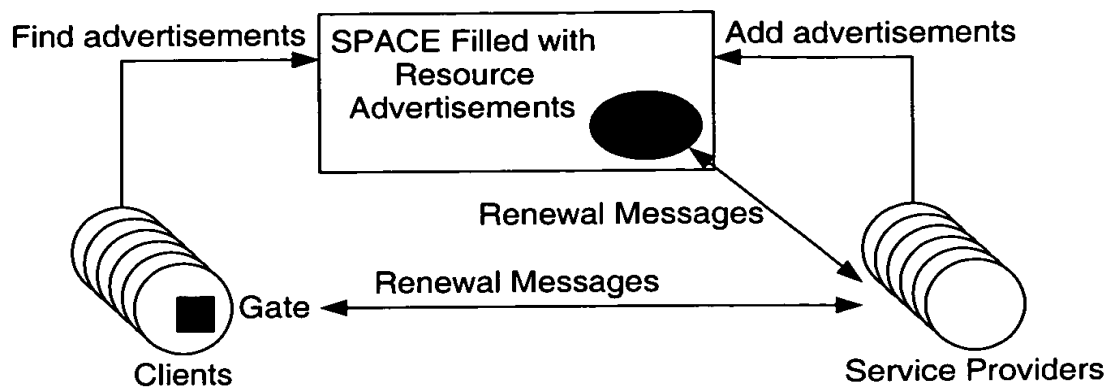


Fig. 25

006T0T 262E6960

SECRET

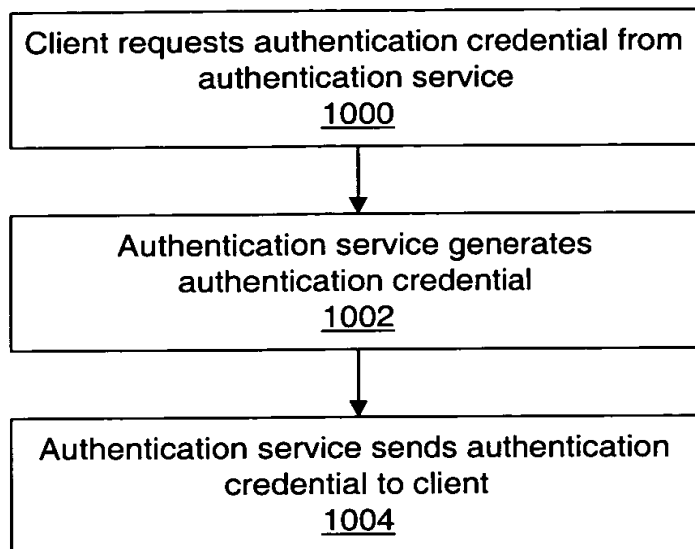


FIG. 26a

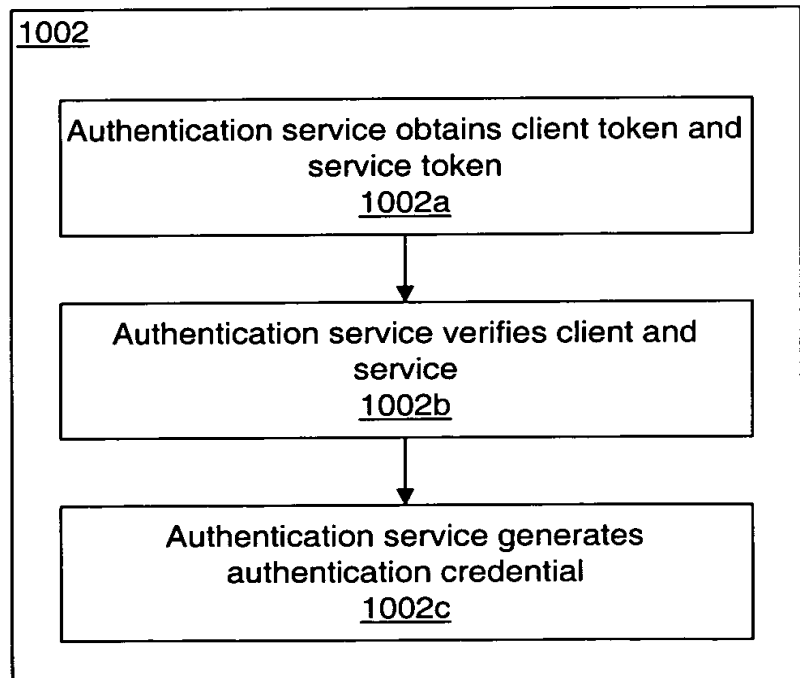


FIG. 26b

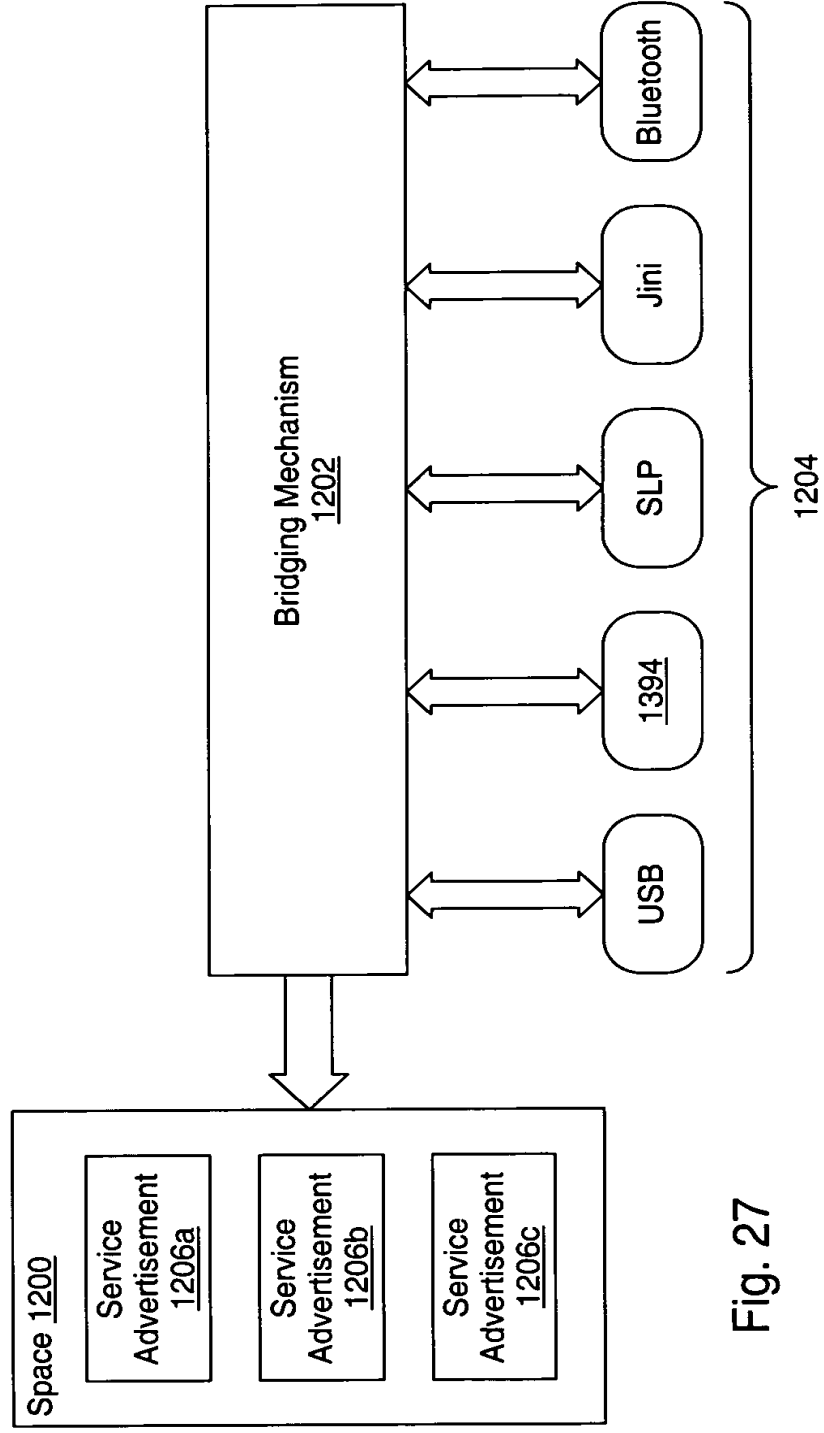


Fig. 27

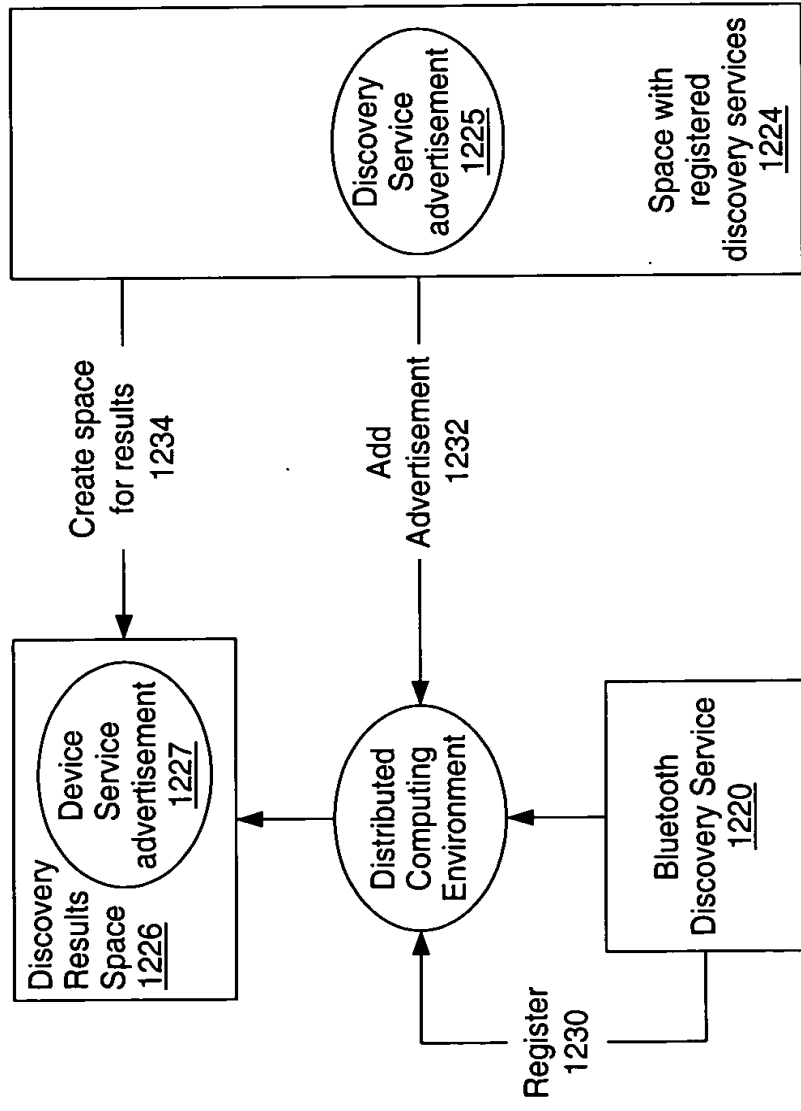


Fig. 28

000001-262E6960

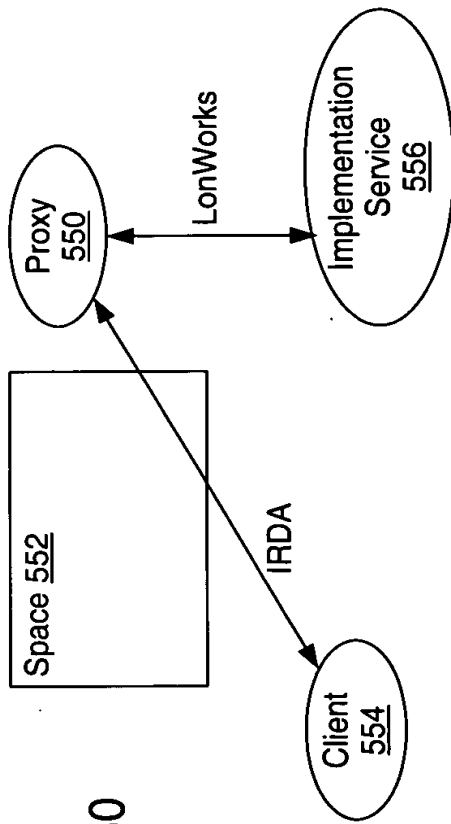


Fig. 30

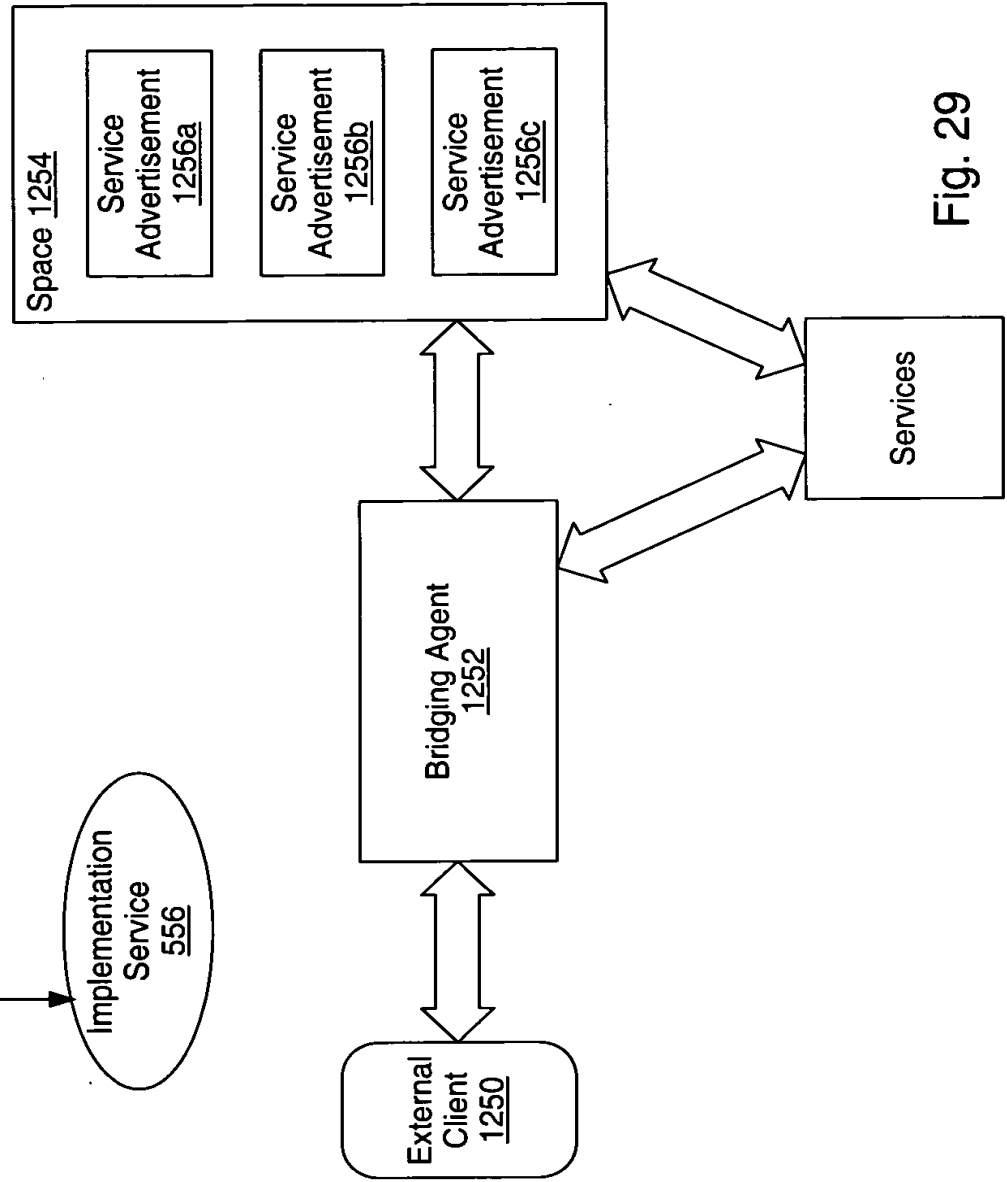
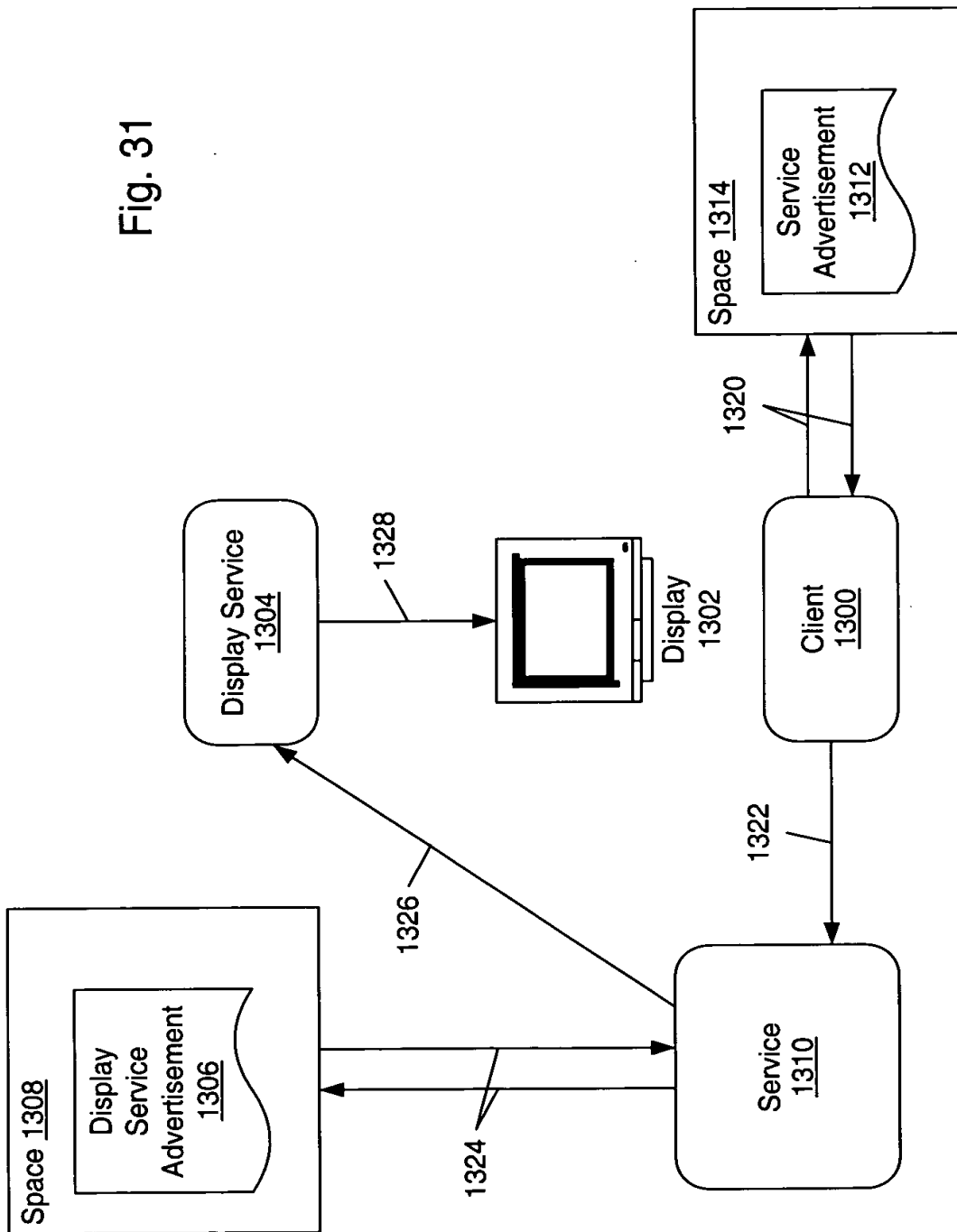


Fig. 29



0066707-26226960

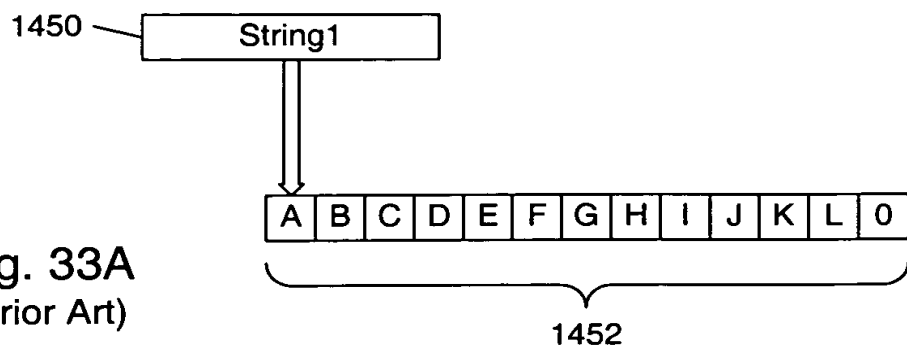


Fig. 33A
(Prior Art)

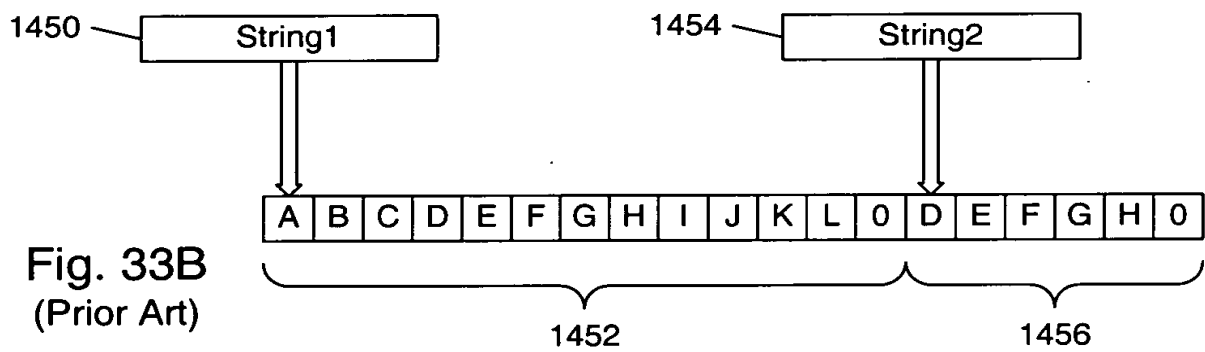


Fig. 33B
(Prior Art)

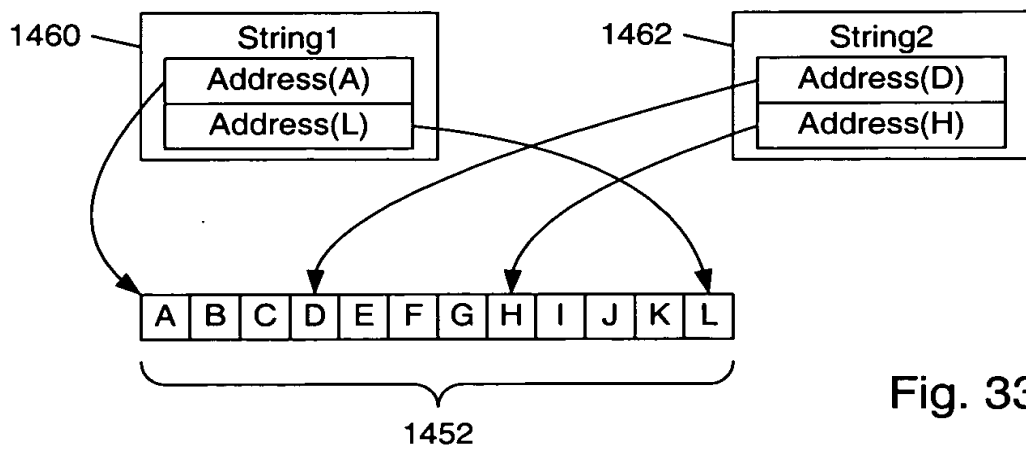


Fig. 33C

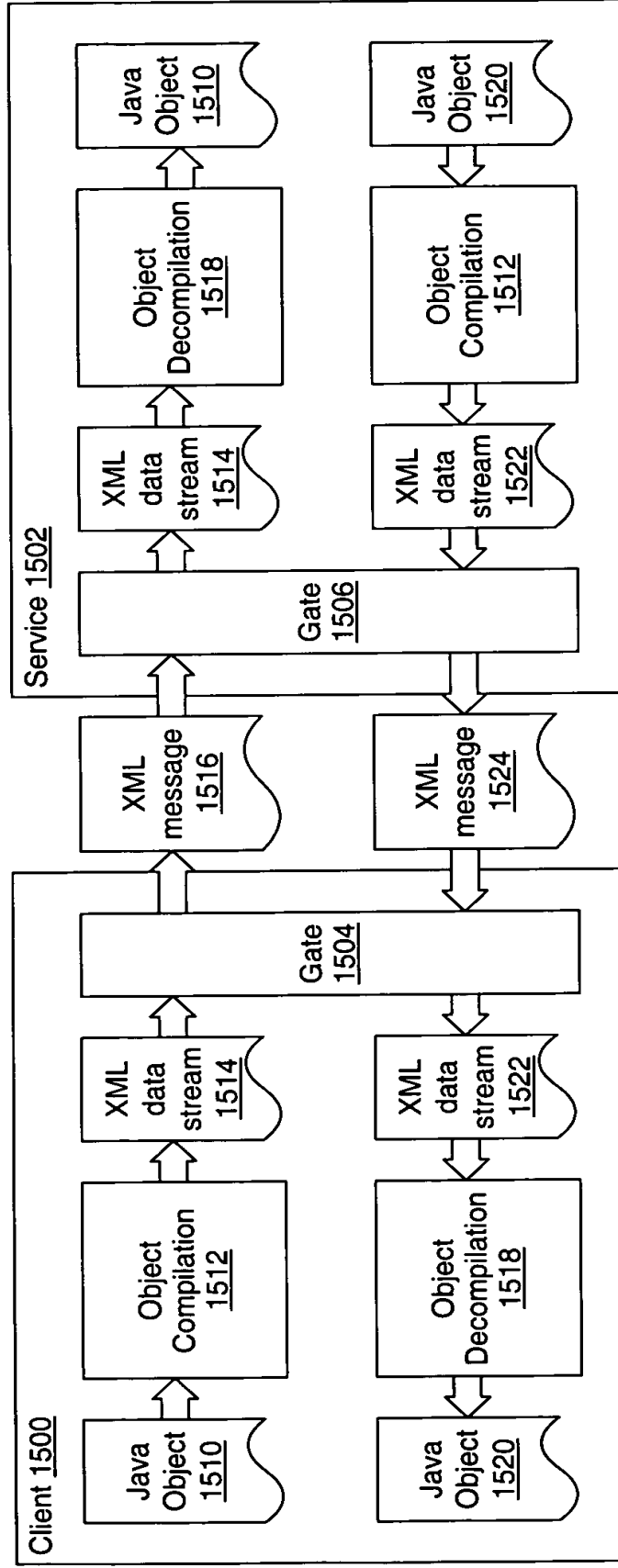


Fig. 34

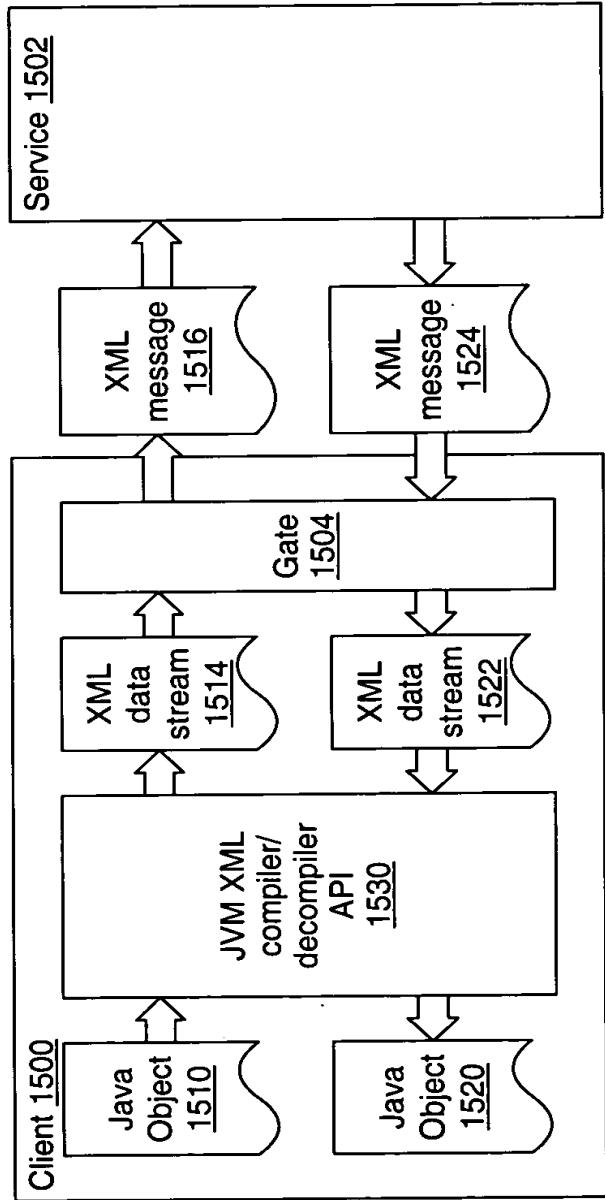


Fig. 35a

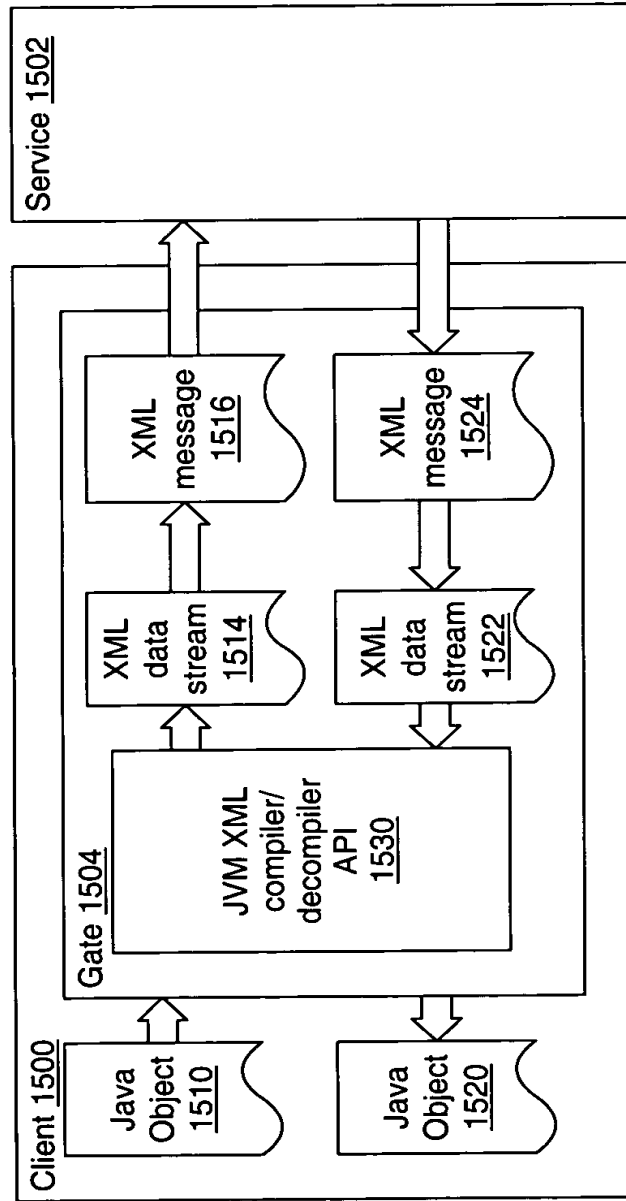
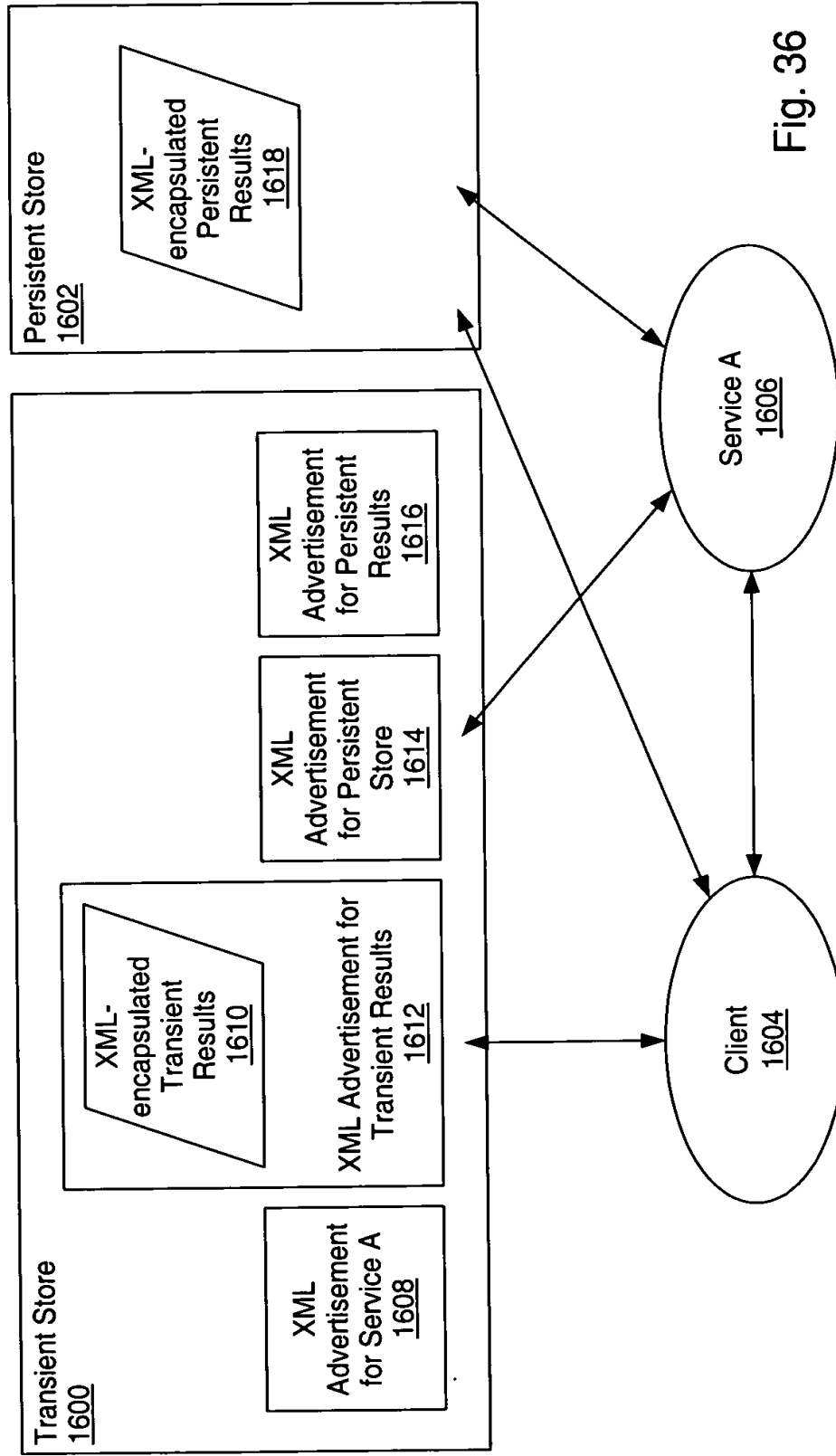


Fig. 35b



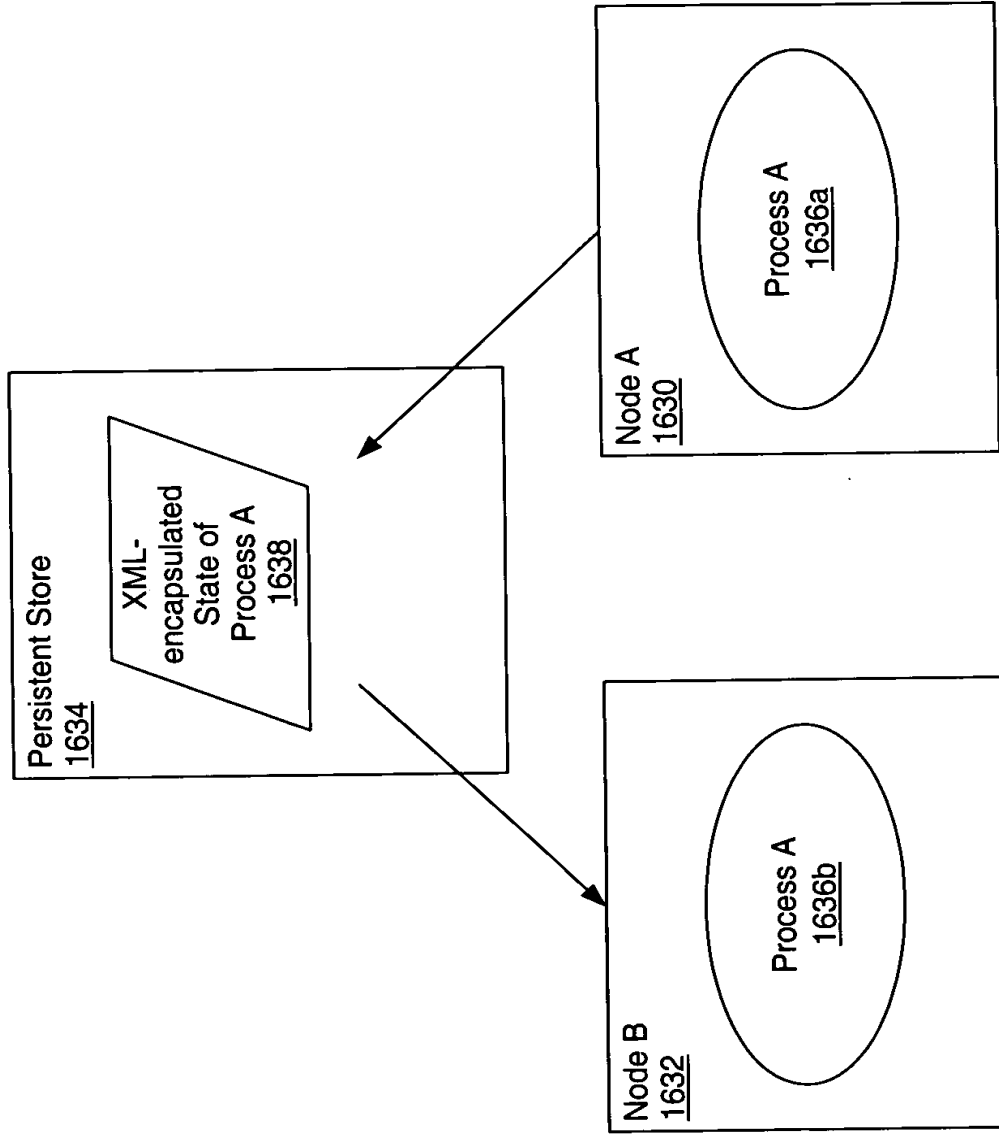


Fig. 37

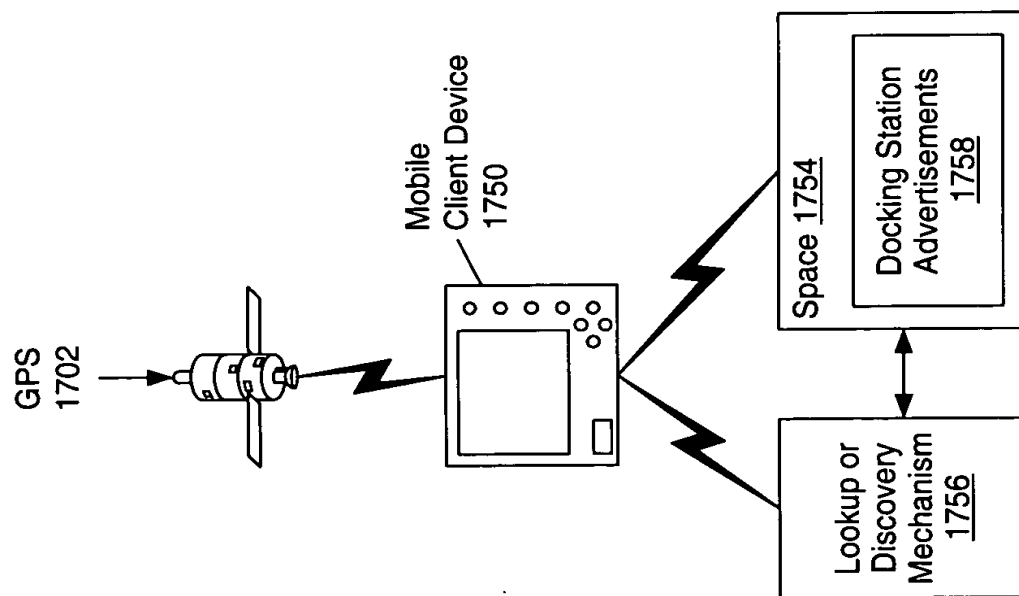


Fig. 39a

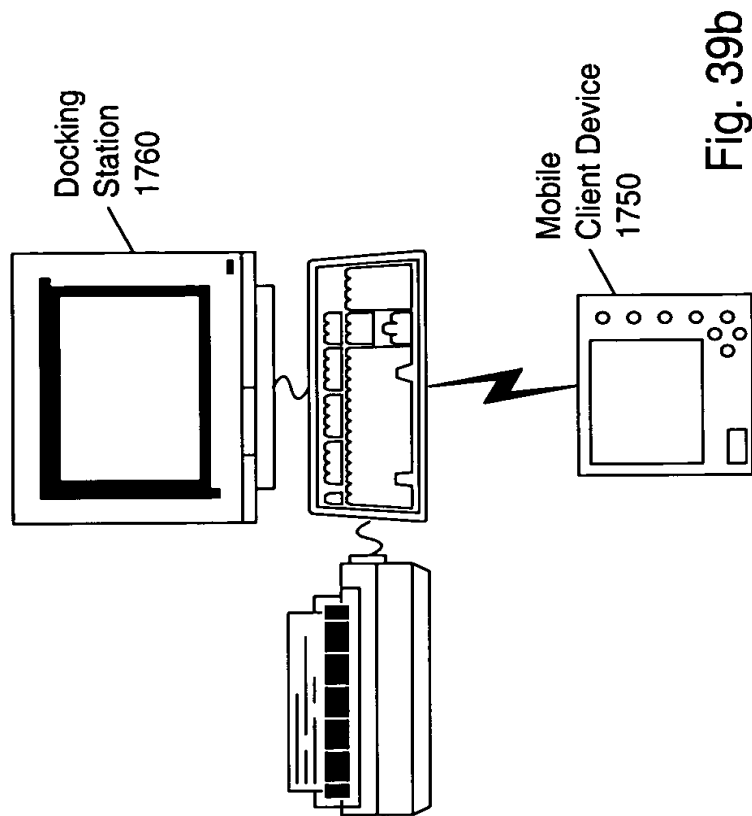


Fig. 39b

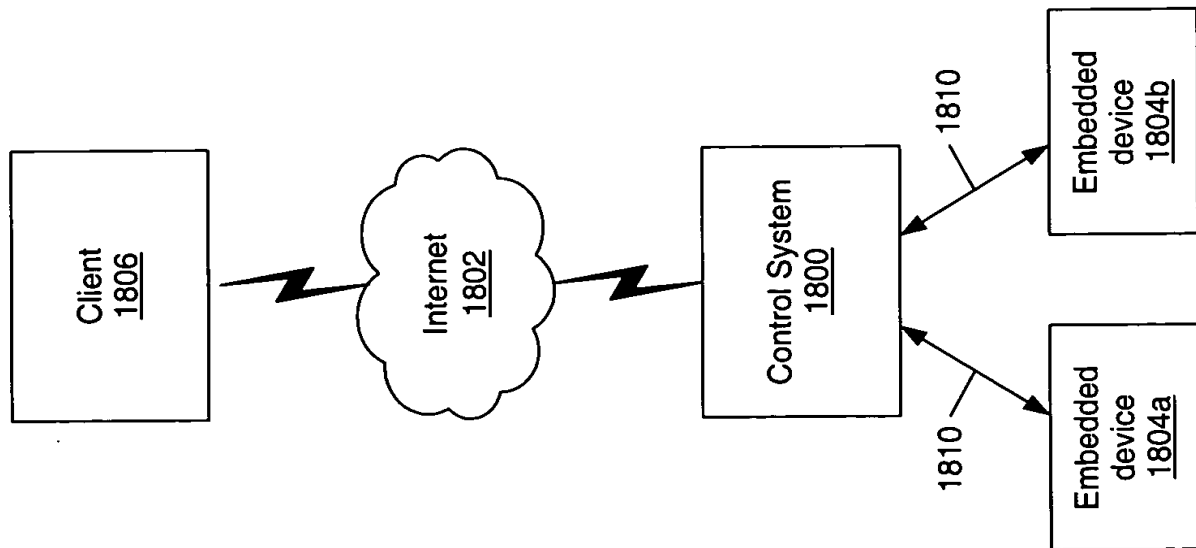


Fig. 40a

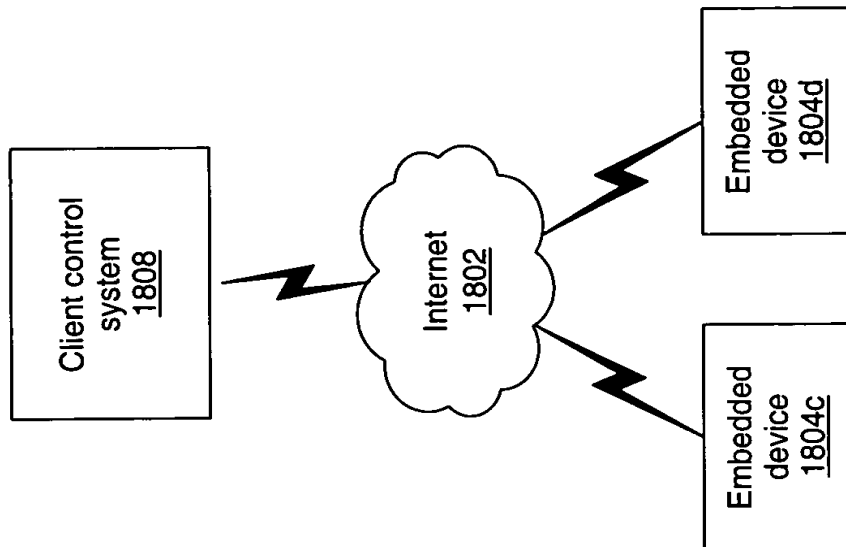


Fig. 40b

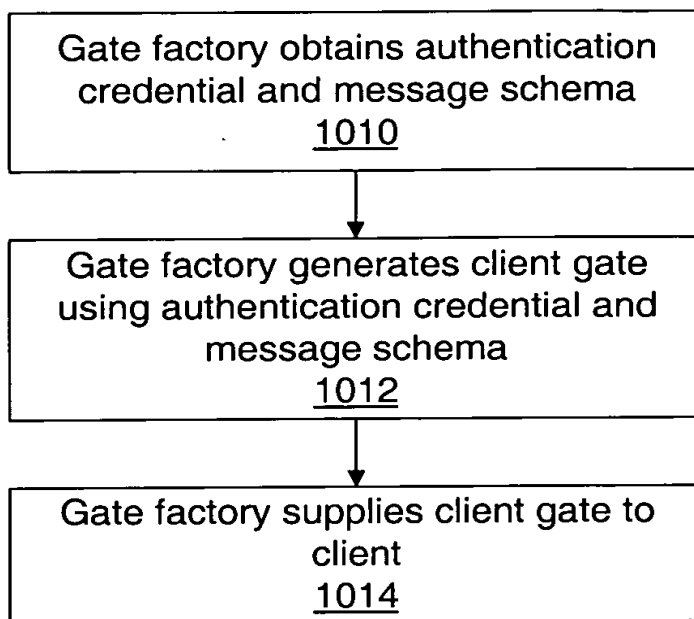


FIG. 41

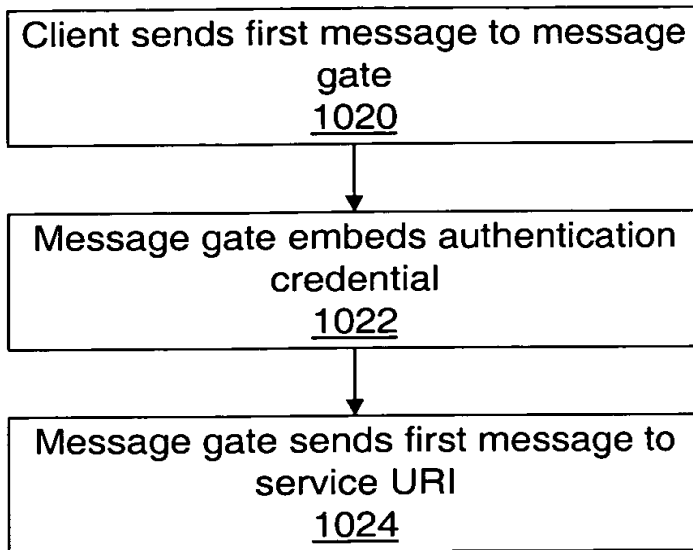


FIG. 42a

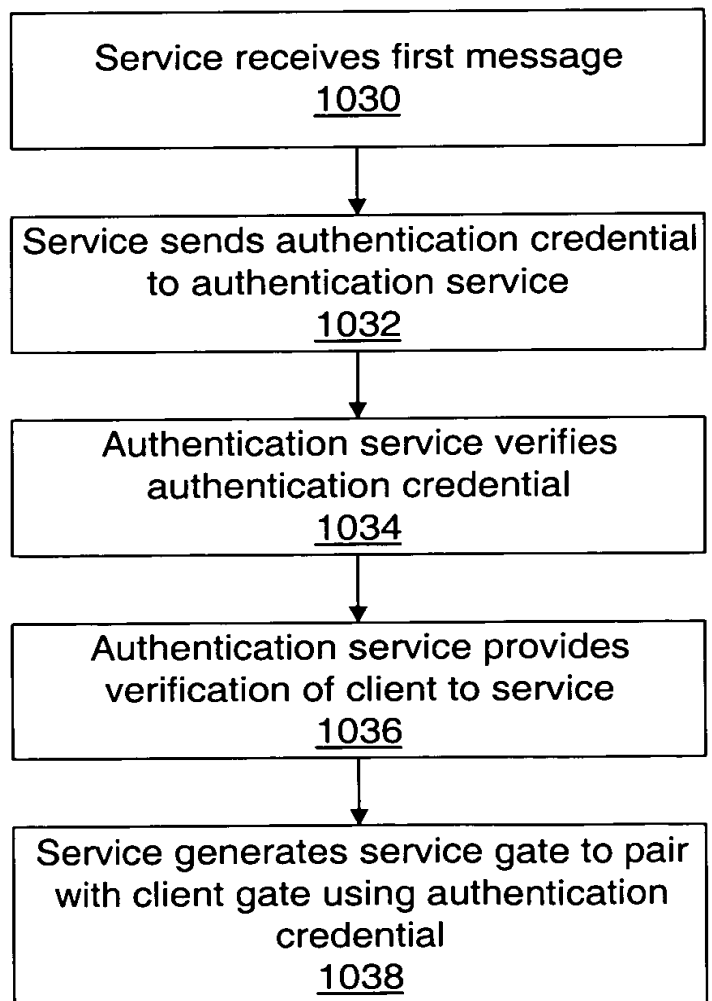


FIG. 42b

006101-101900

006T0T 262E6960

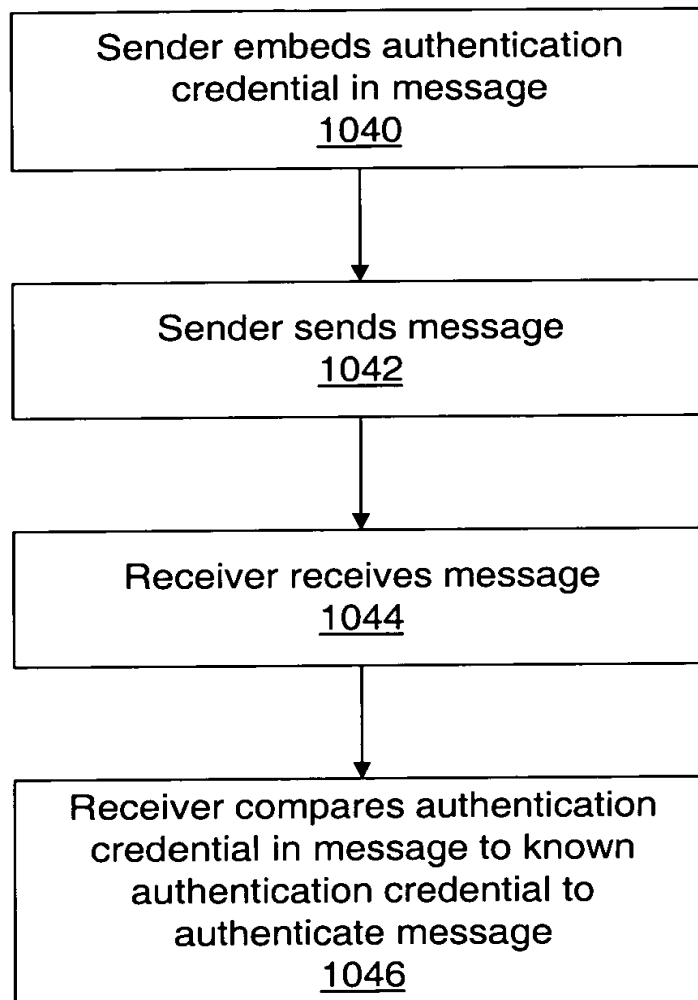


FIG. 42c

```

graph TD
    1050[Sender embeds credential in message  
1050] --> 1052[Sender sends message  
1052]
    1052 --> 1054[Receiver receives message  
1054]
    1054 --> 1056[Receiver examines credential and  
message to verify message has not  
been compromised  
1056]

```

FIG. 43

006101: 20060900

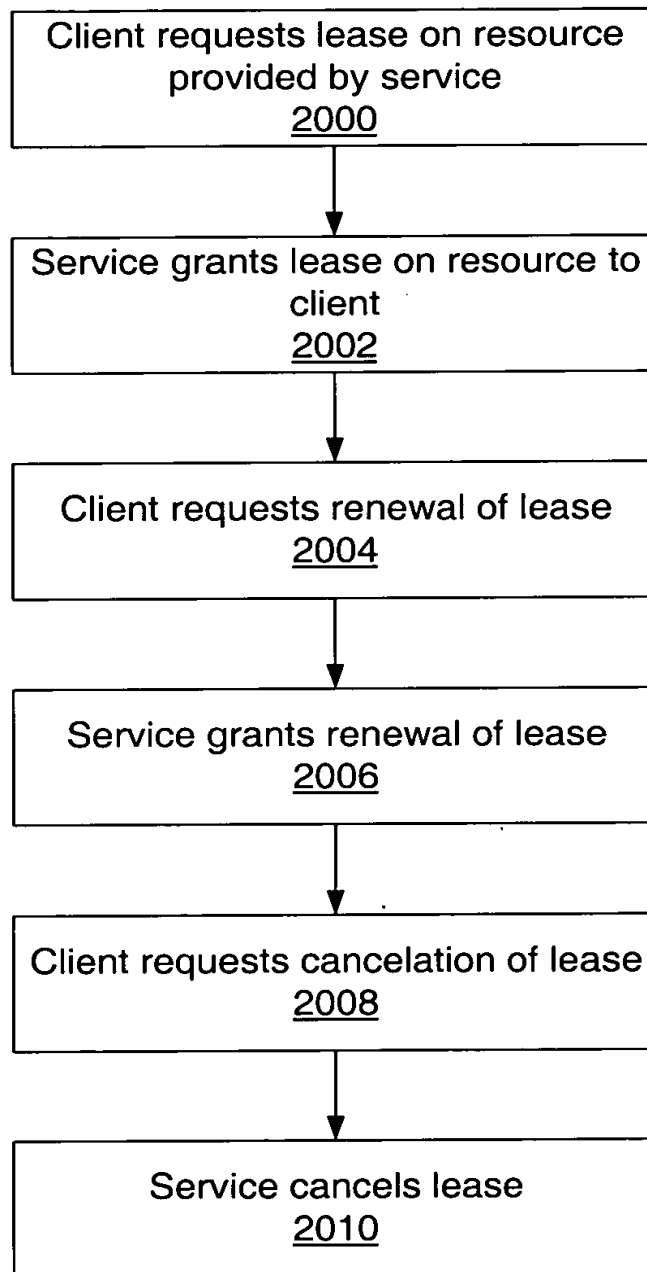


FIG. 44

006707-262E6960

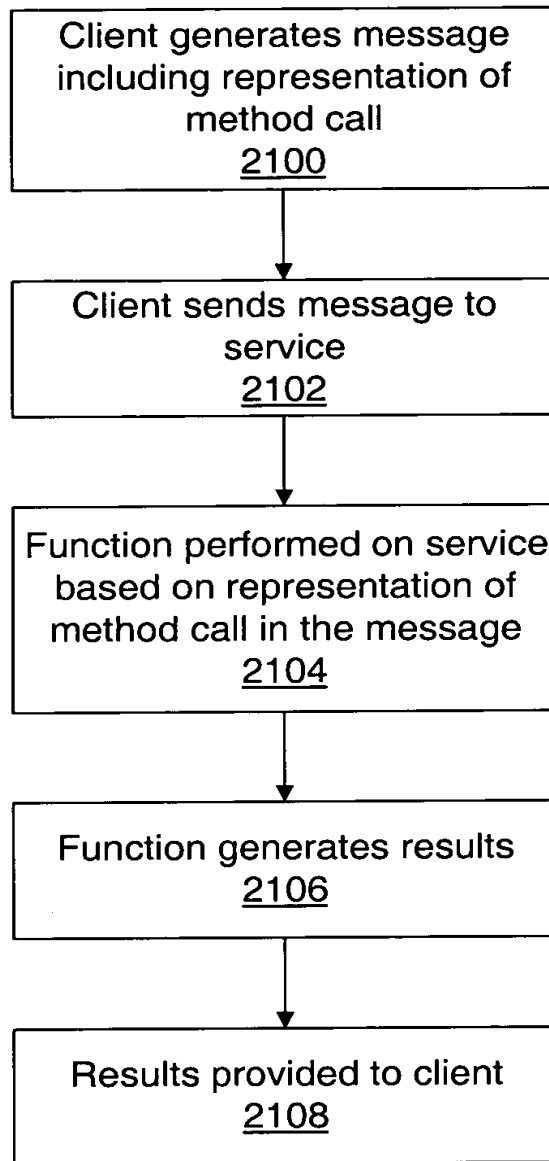


Figure 45a

```

graph TD
    2110[Client makes method call  
2110] --> 2112[Client method gate marshals  
method call into message  
2112]
    2112 --> 2114[Client method gate sends  
message to service  
2114]
    2114 --> 2116[Service method gate  
unmarshals method call from  
message  
2116]
    2116 --> 2118[Method implemented from  
method call  
2118]
    2118 --> 2120[Method generates results  
2120]
    2120 --> 2122[Method results provided to  
client  
2122]

```

```
graph TD; A[Client method gate sends message to service method gate 2114] --> B[Client method gate provides message to client message gate 2124]; B --> C[Client message gate sends message to service message gate 2126]; C --> D[Service message gate sends message to service method gate 2128];
```

Client method gate sends message to service method gate 2114

Client method gate provides message to client message gate 2124

Client message gate sends message to service message gate 2126

Service message gate sends message to service method gate 2128

Figure 45c

```
graph TD; 2122([Method results provided to client  
2122]) --> 2130[Service method gate provides results reference (e.g. advertisement) to service message gate  
2130]; 2130 --> 2132[Service message gate sends results reference to client message gate  
2132]; 2132 --> 2134[Client creates results gate from results reference message  
2134]; 2134 --> 2136[Results gate accesses results on behalf of client  
2136];
```

Method results provided to client
2122

Service method gate provides results reference (e.g. advertisement) to service message gate
2130

Service message gate sends results reference to client message gate
2132

Client creates results gate from results reference message
2134

Results gate accesses results on behalf of client
2136

```

graph TD
    A([Method results provided to client  
2122]) --> B[Service creates results gate  
2140]
    B --> C[Service sends results gate to client  
2142]
    C --> D[Results gate accesses results on behalf of client  
2144]

```

Figure 45e

00692297-101900

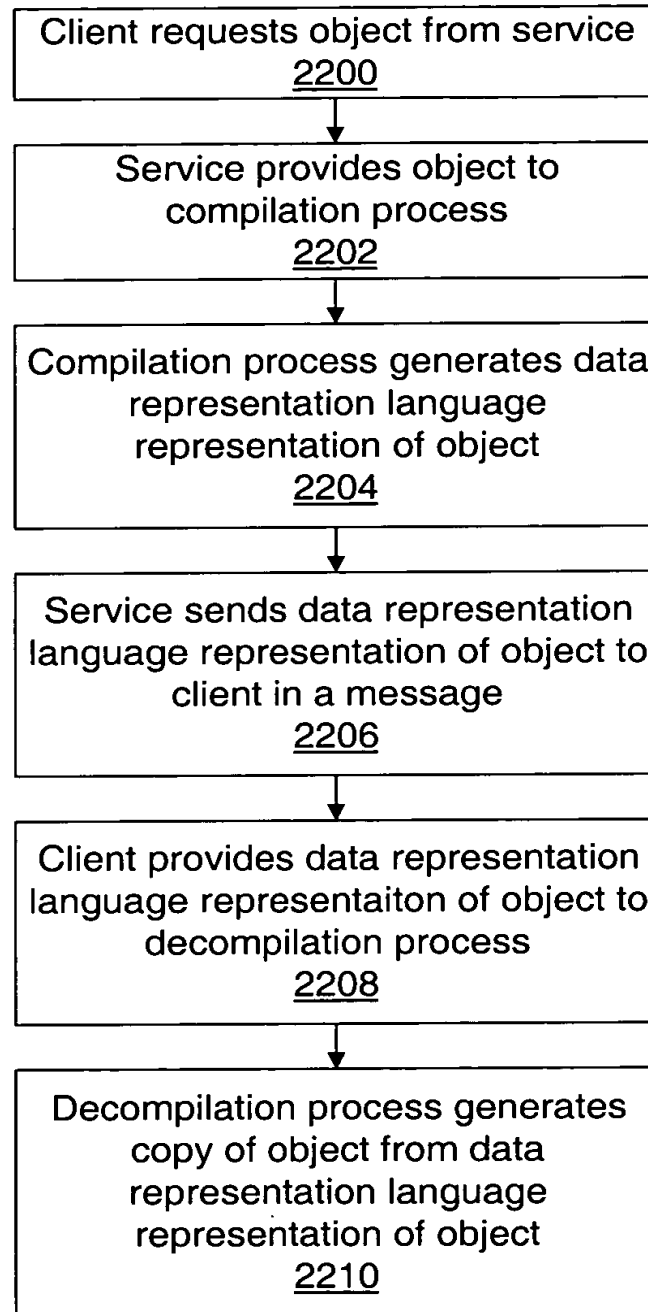


Figure 46

```

graph TD
    2240[User access client  
2240] --> 2242[Decompilation process  
generates copies of objects  
from data representation  
language representations of  
the objects  
2242]
    2242 --> 2244[User ends access of client  
2244]
    2244 --> 2246[Generated objects are  
automatically deleted  
2246]

```

Figure 47


```

graph TD
    2260[Process executes within a first device  
2260] --> 2262[Convert current computation state to a data  
representation language representation of  
the state  
2262]
    2262 --> 2264[Store the data representation language  
representation of the current computation  
state  
2264]
    2264 --> 2266[Second device accesses the data  
representation language representation of  
the current computation state of the process  
2266]
    2266 --> 2268[Reconstitute the process at the current state  
on the second device  
2268]

```

Figure 48

09503297 101400

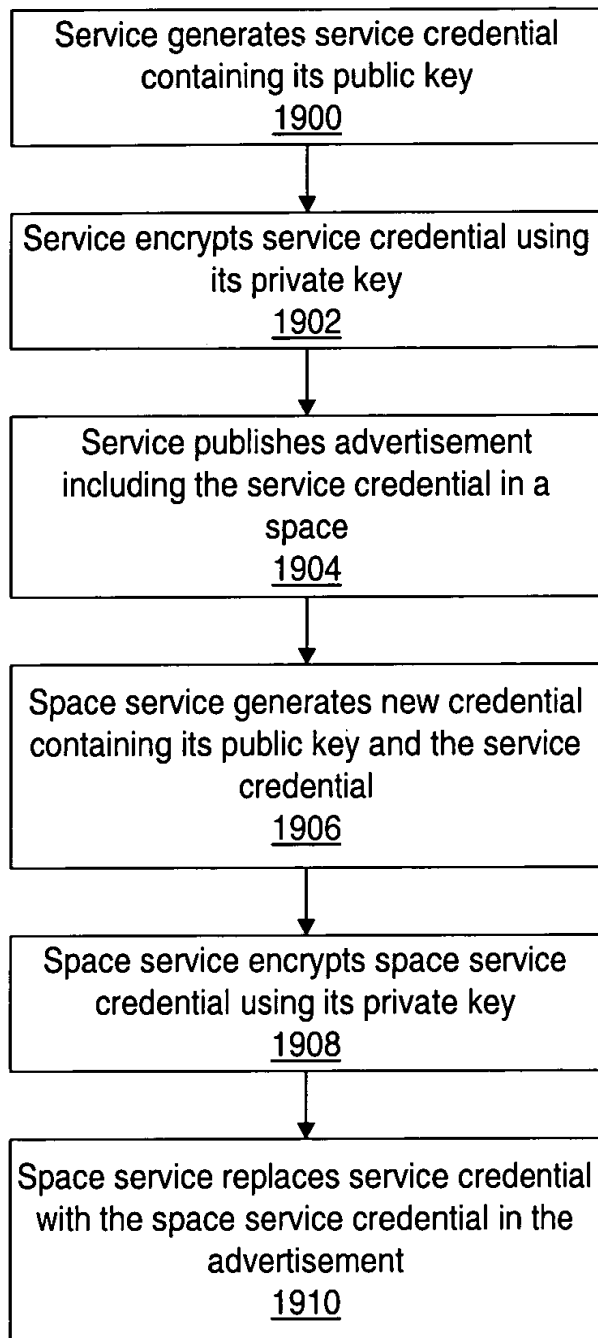


Fig. 49a

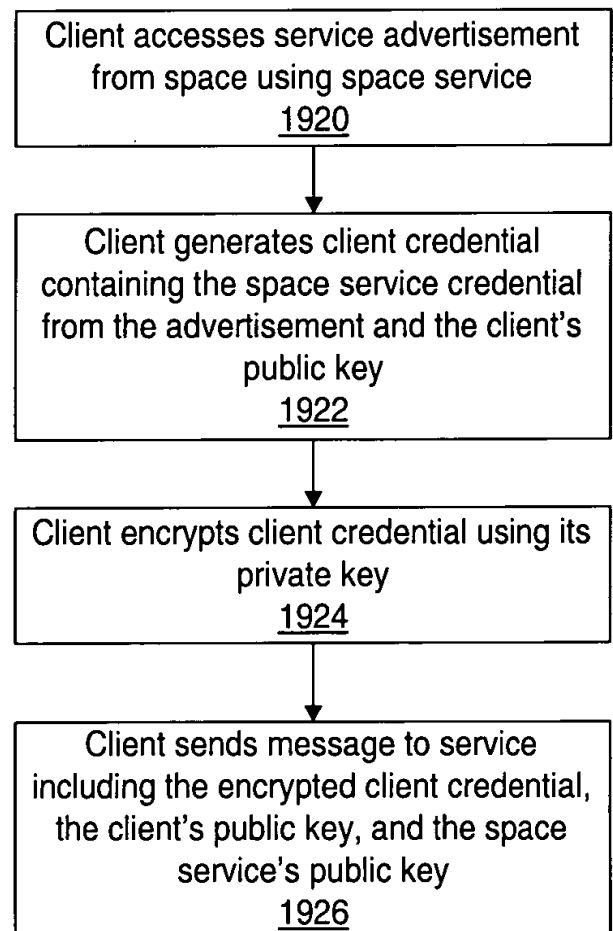


Fig. 49b

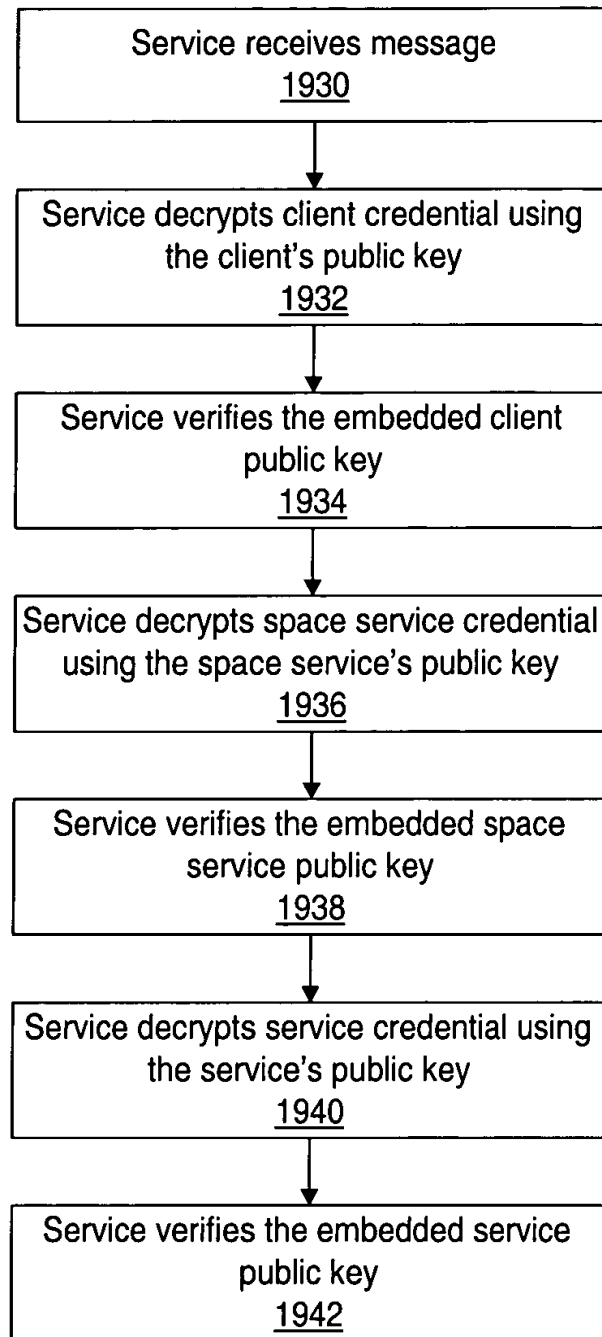



Fig. 49c

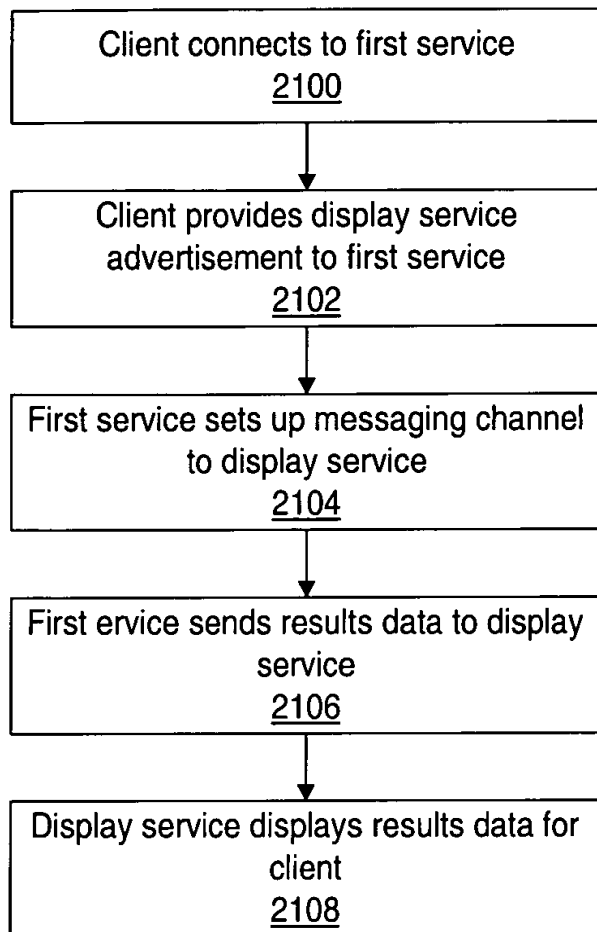


Fig. 50

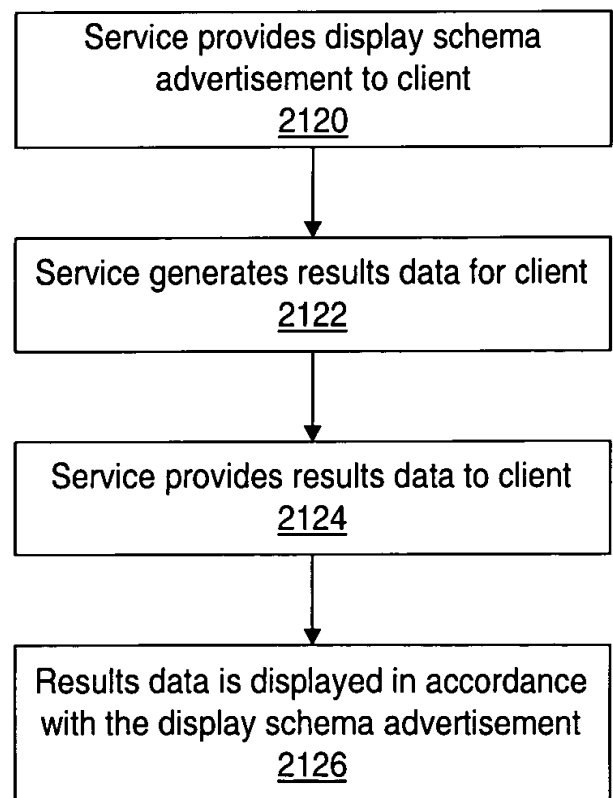


Fig. 51